

UNIVERSITÉ PARIS DIDEROT – PARIS 7

LABORATOIRE PPS

École Doctorale Sciences Mathématiques de Paris Centre

LA PLEINE PARESSE,
UNE CERTAINE OPTIMALITÉ

PARTAGE DE SOUS-TERMES
ET
STRATÉGIES DE RÉDUCTION
EN
RÉCRITURE D'ORDRE SUPÉRIEUR

Thibaut BALABONSKI

Thèse de doctorat
Spécialité informatique

Soutenue le vendredi 16 novembre 2012
devant le jury composé de :

Mme	Zena ARIOLA	
M.	Olivier DANVY	<i>Rapporteur</i>
M.	Stefano GUERRINI	<i>Rapporteur</i>
Mme	Delia KESNER	<i>Directrice</i>
M.	Jean-Jacques LÉVY	
M.	François POTTIER	
M.	Vincent VAN OOSTROM	

Merci

Pour être arrivé jusqu'ici entier, je suis redevable à pas mal de monde.

À Delia Kesner, qui m'a guidé et soutenu pendant plus de quatre ans. À Olivier Danvy, qui m'a permis de comprendre tant de choses. À Vincent van Oostrom, qui m'a offert des intuitions décisives. À Stefano Guerrini, qui m'a suivi semestre après semestre. À Zena Ariola, relectrice tenace. À François Pottier, qui m'accueille dans le monde d'après. À Jean-Jacques Lévy, qui a créé mon fonds de commerce. À tous les sept également car ils me font l'immense honneur de venir juger ce travail.

À Juliusz Chroboczek, dont les commentaires et conseils sont précieux. À Flávio Leonardo Cavalcanti de Moura et Mauricio Ayala-Rincón, qui m'ont réservé un accueil formidable à Brasilia. À Yann Régis-Gianas, car la science est une fête. À Pierre-Louis Curien et Thomas Ehrhard, qui ont fait de PPS un endroit où chacun peut s'épanouir. À Laurent Regnier, qui est un public formidable. À Luc Maranget, Kristoffer Rose, Zurab Khasidashvili, Paul-André Melliés, Maribel Fernández, Barry Jay, qui m'ont conseillé, ont répondu à mes questions ou m'ont vendu des projets fous. À Emmanuel Haucourt et Éric Goubault, avec qui j'ai fait mes premières armes.

À Daniel Hirschhoff, Pierre Lescanne, Roberto di Cosmo, Patrick Baillot, Alexandre Miquel, Gilles Dowek, Xavier Leroy, qui par leurs cours sont un peu responsables de tout ça. À Jacques Mazoyer, Marianne Delorme, Tom Hirshowitz, Éric Thierry, Vincent Danos, Russ Harmer, qui auraient pu l'être aussi. À messieurs Chateaux et Keller, qui m'ont rendu normalisable. À Martin et Julien, car les concours sont aussi un travail d'équipe.

À tous les acolytes qui ont contribué à faire du bureau 6C10 un endroit aussi chouette qu'il est étroit : à Barbara qui m'a légué un siège avec vue sur la Tour Eiffel, au « grand frère » Fabien, au maître des grenouilles Gabriel, au faussement taciturne Jonas, à Stéphane et son canapé, à Antoine et son jazz. Également aux thésards d'en face ou juste de passage, aux buveurs de thé Marie-Aude et Alexis, à Guillaume, à Flavien, au créateur de robots Kuba, à Nicolas, et à Beniamino. Aussi à Christine car que ferions-nous sans elle ?, et à Odile qui fait tourner la baraque.

Aux potes du jeudi soir, Sylvine, Séb, Basile, Aude, Justine, et tous les altistes, et à ceux des autres jours, Mathilde, Fréd, Cloé, Agostinho, Florence, Olivier, Stéphanie, et Jonathan.

À mes parents qui m'ont donné les moyens d'arriver jusque là.
Et à Mireille, tout simplement.

Thèse :

L'évaluation pleinement paresseuse à la Wadsworth
est une réalisation de la réduction optimale à la Lévy
pour les systèmes de réécriture du second ordre orthogonaux faibles.

Introduction

```
Closure 33, 0. Push. Closurerec 12, 0.
Const [0: 2 [0: 3 [0: 4 0a]]]. Push. Const 1.
    Push. Acc 3.
    Push. Acc 3.
    Appterm 3, 6. Restart.
    Grab 2. Acc 2. Branchifnot 30. Acc 2.
    Getfield 1. Push. Acc 3.
    Getfield 0. Push. Acc 3.
    Push. Acc 3.
Apply 2. Push. Acc 2. Push. Offsetclosure 0.
    Appterm 3, 6. Acc 1. Return 3. Restart.
    Grab 1. Acc 1. Push. Acc 1.
    Mulint. Return 2.
```

Et l'ordinateur répondit :

24.

Le sens des programmes

L'ordinateur est un exécutant parfait. Que des ordres lui soient donnés, il les suit sans sourciller. Il va vite, il ne se trompe pas, il ne connaît pas l'Ennui. Il fait tout ce qui lui est demandé, rien que ce qui lui est demandé, et tout cela sans s'inquiéter des raisons de ses actes.

Et pour cause, l'ordinateur vit dans le *comment*, et ne comprendrait rien au *quoi* ni au *pourquoi*.

Il est parfaitement inutile de lui dire « j'ai une liste de nombres et j'aimerais en connaître le produit ». Pour le faire bouger, tout ce qui compte est de lui préciser la manière dont il doit s'y prendre et lui détailler pas à pas tout ce qu'il doit faire, en n'utilisant que les quelques dizaines d'*instructions* de base qu'il comprend, comme multiplier deux nombres entiers ou aller rechercher en mémoire une information préalablement enregistrée.

Chaque ordinateur a ainsi son propre *langage machine*, formé d'une poignée de mots et de la plus pauvre des grammaires, dont les phrases permettent seulement de lui affecter les quelques tâches qu'il sait réaliser. Pour utiliser ce langage, il faut connaître parfaitement les mécanismes de l'ordinateur, et être

capable sans le secours d'Ornicar de prévoir en détail les effets d'une jungle de monosyllabes.

Avouons que programmer dans ces conditions est laborieux. Pour simplifier cela nous utilisons des *langages de programmation* au vocabulaire et aux structures grammaticales plus riches. Nous nous intéresserons particulièrement aux langages dit *déclaratifs*, dans lesquels la description du *quoi* remplace partiellement celle du *comment*, par exemple par l'utilisation d'équations mathématiques.

```

mult(x, y) = x * y
fold(f, z, [] ) = z
fold(f, z, x:xs) = fold(f, f(z,x), xs)

fold(mult, 1, 2:3:4:[])

```

Le sens d'un tel programme de haut niveau n'est pas aussi explicite que ne l'était celui d'un programme machine : ici il n'y a plus d'instructions précises et univoques. En revanche, nous pouvons utiliser les trois équations données dans les premières lignes du programme précédent pour en transformer la dernière afin de *calculer* son résultat. Ce calcul est montré ci-dessous, et à chaque étape est souligné l'opérateur principal sur lequel une équation est appliquée. L'expression remplacée à l'aide d'une équation est appelée un *radical*¹.

```

fold(mult, 1, 2:3:4:[]) = fold(mult, mult(1,2), 3:4:[])
                        = fold(mult, 2, 3:4:[])
                        = fold(mult, mult(2,3), 4:[])
                        = fold(mult, 6, 4:[])
                        = fold(mult, mult(6,4), [])
                        = fold(mult, 24, [])
                        = 24

```

Mais gardons à l'esprit la finalité de notre programme : être exécuté par un ordinateur, non dans un monde d'abstraction mathématique mais dans le monde réel qui en dernier recours reste la source du droit, et où se trouvent la motivation originelle et la validation ultime des travaux dans lesquels s'inscrit cette thèse.

La situation est alors la suivante : nous avons un programme écrit dans un langage déclaratif relativement clair quant au *quoi* mais un peu flou sur le *comment*, et nous voulons le traduire dans un langage machine qui nécessite une grande précision dans le *comment*. Il va donc falloir lever les zones d'ombre, et pour cela choisir une voie parmi la multitude des possibilités ouvertes dans le monde de la manipulation d'équations.

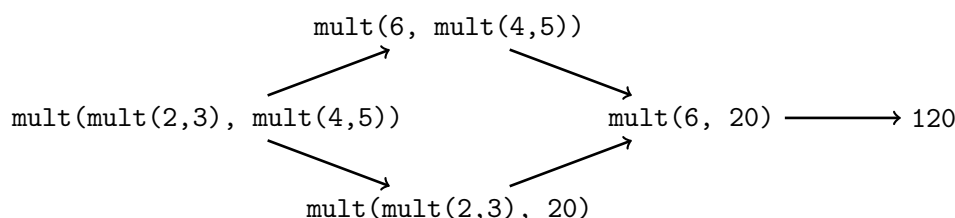
Choisir, avec discernement bien sûr.

1. Comme en chimie, ce mot désigne une entité prête à réagir. En anglais la même idée transparaît avec le mot *redex* qui est une contraction de *reducible expression*.

Stratégies de calcul

Pour étudier et comparer les multiples ordres dans lesquels peuvent être effectuées les opérations décrites par un programme déclaratif, nous allons prendre comme unité de référence l'action de remplacer (ou *récrire*) un fragment de programme en se servant d'une équation. Pour symboliser le fait que ces remplacements ont chacun un « avant » et un « après », deux étapes consécutives du calcul seront liées par une flèche \rightarrow plutôt que par un signe d'égalité.

À chaque programme nous pouvons associer une carte des calculs possibles, où figurent tous les états intermédiaires envisageables et où des flèches marquent les pas de calcul (ou *pas de réduction*) permis. Un calcul est un cheminement à travers cette carte appelée *espace de réduction*.



L'enjeu ici est la connaissance de stratégies permettant d'atteindre le résultat d'un programme en empruntant un chemin de calcul parmi les plus courts possibles et en évitant deux écueils :

- D'abord, il arrive que certains pas de calcul intermédiaires ne soient pas nécessaires à l'obtention du résultat. Prenons par exemple l'expression

$$\text{fold}(f, 1, [])$$

Nous savons sans regarder f que cette expression est égale à

$$1$$

Quelle que soit la forme de l'expression f il n'est donc pas utile de calculer son résultat.

- Ensuite, certains pas de calcul peuvent avoir comme effet secondaire de multiplier le travail restant à effectuer. Prenons par exemple l'expression

$$\text{fold}(f, 1, 2:3:[])$$

Nous pouvons effectuer un pas de calcul pour atteindre

$$\text{fold}(f, f(1,2), 3:[])$$

où l'expression f apparaît deux fois (contre une dans l'expression d'origine). Bilan : s'il y a encore du calcul à faire dans f , alors nous serons peut-être amenés à le faire deux fois.

La prochaine double-page contient des exemples détaillés de ces deux phénomènes : la non-nécessité à la figure 1 et les duplications à la figure 2.

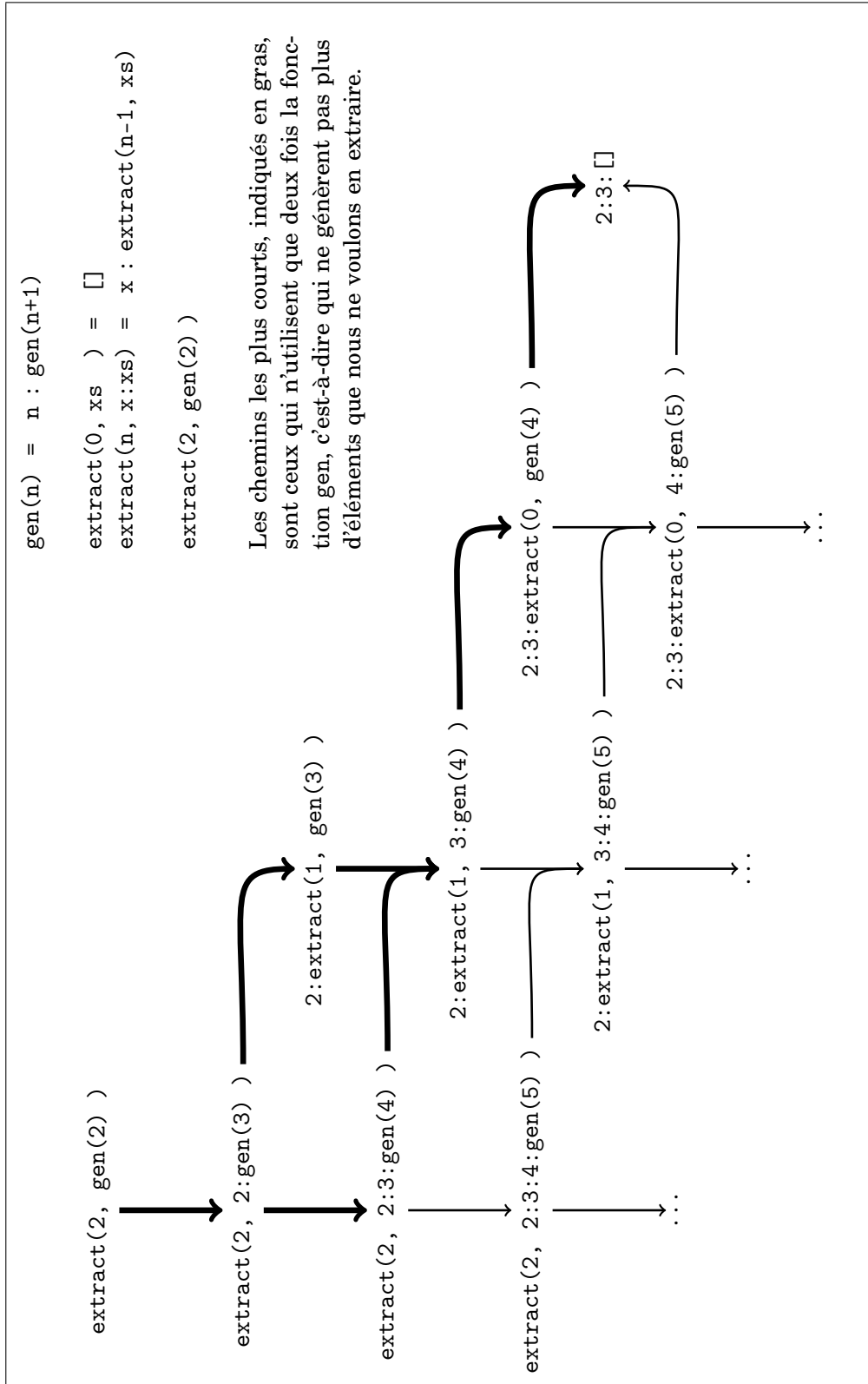


FIGURE 1: Non-nécessité

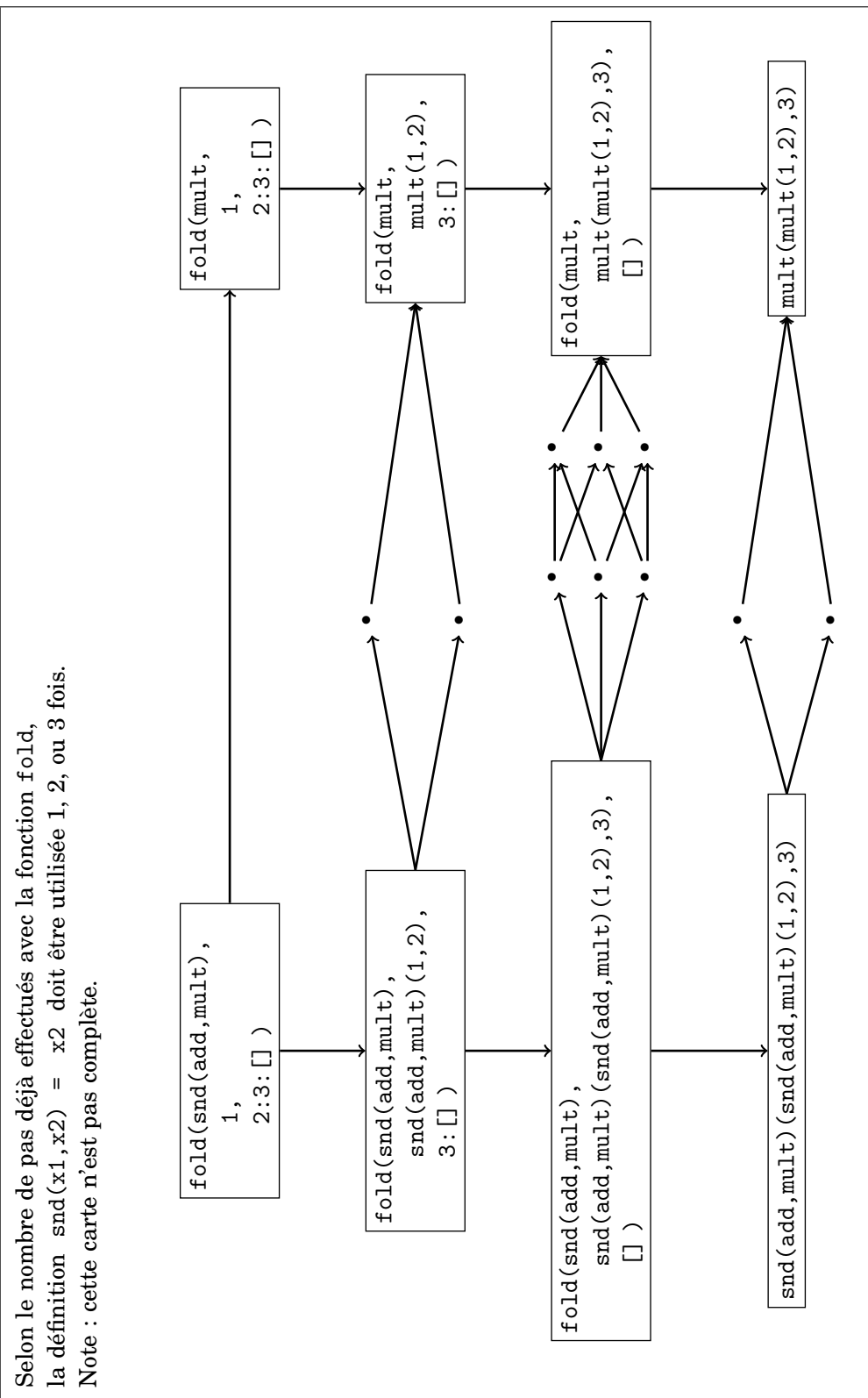


FIGURE 2: Duplications

Le λ -calcul

De nombreux travaux se sont déjà penchés sur la question des stratégies de calcul en ce concentrant sur le noyau fondamental de la programmation dite *fonctionnelle* connu sous le nom de λ -calcul. Il s'agit d'un modèle minimaliste mais néanmoins expressif, centré sur une unique opération : l'application d'une fonction à ses arguments.

Un exemple typique de fonction s'exprimant directement en λ -calcul est la fonction `comp` suivante qui, lorsque lui sont donnés en arguments deux fonctions (notées `f` et `g`) et un élément (noté `x`), applique successivement les deux fonctions à l'élément :

$$\text{comp}(f, g, x) = g(f(x))$$

Le λ -calcul offre une notation compacte de ce comportement en représentant l'application `f(x)` par la simple juxtaposition `fx` de `f` et `x`, et en indiquant avec le symbole λ les noms des *variables* d'une fonction.

$$\text{comp} = \lambda f. \lambda g. \lambda x. g(fx)$$

Le mécanisme de calcul peut agir quand une fonction est appliquée à un argument, par l'action de réécriture qui remplace toutes les occurrences de la première variable de la fonction par l'argument. Ainsi par exemple :

$$(\lambda f. \lambda g. \lambda x. g(fx))(\lambda y. yy) \rightarrow \lambda g. \lambda x. g((\lambda y. yy)x)$$

Cette opération de substitution a une forme générale notée

$$(\lambda x. t)a \rightarrow t^{x:=a}$$

qui à elle seule définit tout ce modèle abstrait, et contient nos deux problèmes de non-nécessité et de duplication : l'expression `a` va apparaître dans $t^{x:=a}$ autant de fois que la variable `x` est utilisée dans `t`, ce qui peut faire disparaître `a` aussi bien que le dupliquer.

La tension se trouve donc dans la question de l'évaluation de l'expression `a`, avec deux stratégies simples emblématiques des deux faces de la question.

- *L'appel par valeur* ne passe un argument à une fonction qu'après l'avoir évalué. Dans une expression comme $(\lambda x. t)a$ la première tâche accomplie est le calcul de la valeur `v` de l'argument `a`, après quoi la substitution $t^{x:=v}$ est effectuée. Alors l'argument `a` est toujours évalué exactement une fois, que son résultat serve une fois, dix fois, ou jamais.
- *L'appel par nom* effectue directement la substitution et remet à plus tard l'éventuelle évaluation de l'argument. De cette manière l'argument `a` est évalué autant de fois que son résultat est demandé.

Notre question « étant donnée une expression, savons-nous atteindre son résultat en empruntant un chemin parmi les plus courts de son espace de réduction ? » est donc très vive dans le λ -calcul, et un certain nombre de jalons ont déjà été posés depuis les années 70.

Ignorabimus

La science n'est pas omnipotente. C'est comme ça. Les mathématiques et l'informatique, en tant qu'objets abstraits de facture humaine, ont cependant un avantage : nous avons quelques moyens de raisonner formellement sur leurs limites, et d'assurer que certaines questions leur sont et leur resteront à jamais hors de portée.

Ainsi en est-il de la stratégie *optimale* du λ -calcul. Étant donnée une expression t du λ -calcul et son espace de réduction, et en admettant que cet espace de réduction contienne une expression t' qui soit le résultat de l'évaluation de t , il est évident qu'il existe un chemin le plus court de t vers t' . Mais Barendregt, Bergstra, Klop et Volken ont démontré [BBKV76, Bar84] qu'il n'était pas possible d'emprunter ces chemins *à coup sûr*. Autrement dit il n'existe pas de méthode automatique, et il n'est pas possible qu'il existe une méthode automatique, permettant pour chaque expression du λ -calcul de savoir quel est le prochain pas de calcul à effectuer pour rester sur le chemin le plus court vers l'éventuel résultat.

Barendregt, Bergstra, Klop et Volken ont donc démontré qu'il était possible d'avoir du travail à vie : la meilleure stratégie ne nous étant pas accessible, l'espoir de pouvoir améliorer les techniques existantes sera toujours d'actualité. Également, cette limitation est une bonne motivation pour décaler légèrement son point de vue et essayer d'autres angles pour aborder la question.

L'appel par nécessité à la Wadsworth

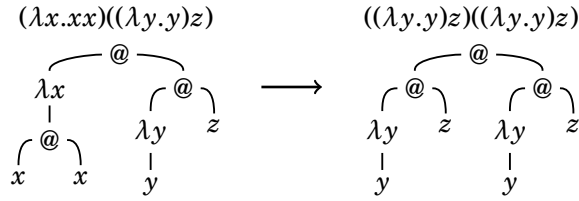
En 1971, Wadsworth a le premier résolu la tension entre l'appel par valeur et l'appel par nom, en proposant une nouvelle technique pour l'évaluation du λ -calcul qu'il a baptisée *appel par nécessité* [Wad71], et dont les diverses émanations ont encore été nommées évaluation paresseuse et évaluation pleinement paresseuse par la suite.

Comme le suggère son nom, cette technique n'effectue que des pas de calcul dont nous savons qu'ils sont nécessaires à l'obtention d'un résultat. Elle cible toujours pour cela le pas de calcul défini par le symbole λ le plus à gauche dans l'écriture de l'expression, dont la nécessité ne souffre aucun doute [CF74]. Il s'agit d'un choix usuel parmi les stratégies n'effectuant que des pas de calcul nécessaires. En outre, et il s'agit du point délicat, cette technique fait en sorte de ne jamais évaluer deux fois une expression qui n'aurait été évaluée qu'une seule fois en appel par valeur.

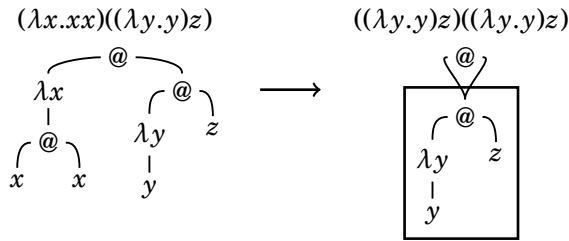
Ceci n'est pas possible avec le seul λ -calcul. Si dans le terme

$$(\lambda x.xx)((\lambda y.y)z)$$

nous effectuons le pas de calcul correspondant au λ le plus à gauche, alors l'expression $(\lambda y.y)z$ se trouve dupliquée sans retour possible, comme le montre la représentation graphique ci-dessous dans laquelle le symbole @ représente l'application d'une expression à une autre.

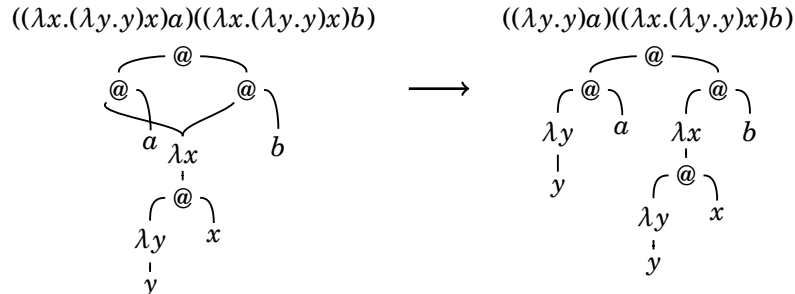


Pour réaliser sa technique, Wadsworth a introduit une nouvelle représentation des expressions du λ -calcul. Nous passons d'une représentation arborescente en *termes* liée à l'écriture textuelle dans laquelle les différentes branches d'une expression sont séparées, à une représentation en *graphes* dans laquelle des branches distinctes peuvent se rejoindre. Ainsi, ci-dessous, alors que l'argument $(\lambda y.y)z$ est *logiquement* dupliqué à cause des deux occurrences de la variables x , nous n'en gardons qu'une seule copie *physique* qui est dite *partagée*. La sous-expression partagée est encadrée.



Passer du monde des termes à celui des graphes permet d'obtenir un nouvel espace de réduction, dans lequel peuvent être faites en une fois des *étapes* de calcul correspondant à plusieurs pas de calcul sur les termes (figure 3). Nous utiliserons une double flèche \Rightarrow pour désigner ces étapes de calcul, en contraste avec la simple flèche \rightarrow utilisée pour les pas de calcul.

Notons que cette technique ne fait que retarder les duplications en espérant en avoir le moins possible à faire en définitive, et ne peut toutes les éviter. Par exemple, une fonction partagée doit être dupliquée avant d'être appliquée, sans quoi nous n'aurions plus sous la main de version « non usagée » de la fonction au moment de sa prochaine application à un argument différent. C'est le cas de la fonction $\lambda x.(\lambda y.y)x$ ci-dessous, dont l'application à l'argument a nécessite la création d'une nouvelle copie de l'expression $(\lambda y.y)x$ dans laquelle la variable x est remplacée par a ².



2. La technique pleinement paresseuse de Wadsworth ne duplique pas intégralement une fonction lors de son application. Ici, nous pourrions garder une unique copie de l'expression $\lambda y.y$. En revanche, les deux radicaux $(\lambda y.y)a$ et $(\lambda y.y)x$ seraient encore séparés par l'opération.

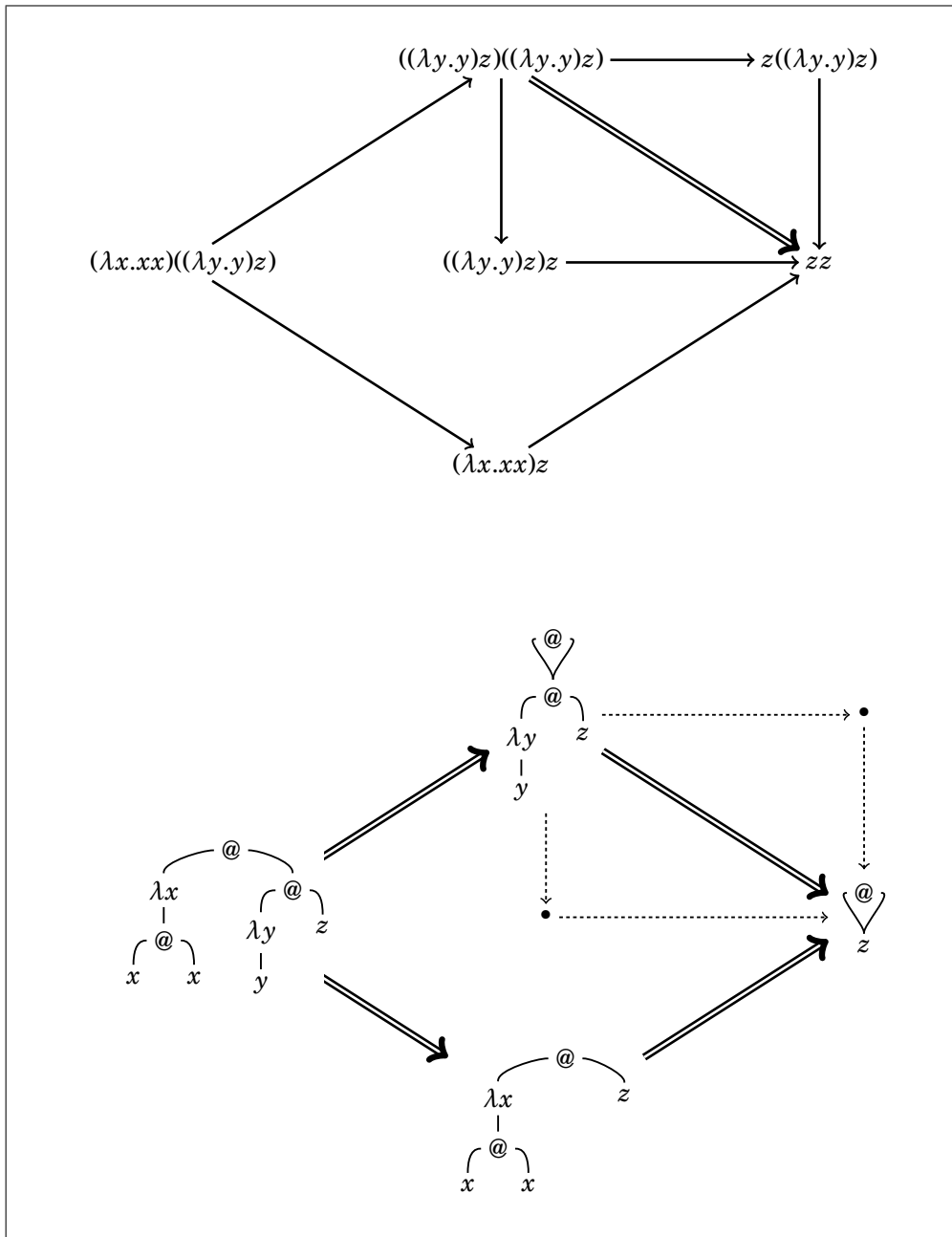


FIGURE 3: Pas de calcul \rightarrow et étapes de calcul \Rightarrow .

L'optimalité à la Lévy

Wadsworth a introduit l'idée du partage en réécriture pour contrer les duplications de travail qui étaient liées à des duplications de termes. Il a matérialisé cette idée dans une technique d'évaluation partagée du λ -calcul utilisant des graphes pour éviter certaines duplications. Cependant, nous avons vu également que la technique de Wadsworth n'élimine pas toutes les duplications. Vient alors la question suivante : « est-il possible de ne rien dupliquer ? » La première difficulté se trouve dans l'interprétation de la question. Avons-nous les idées claires sur ce qu'est une duplication ? Savons-nous à coup sûr quand deux radicaux sont deux duplicatas d'un même ancêtre ?

Regardons la séquence de réduction suivante et en particulier le radical souligné $(\lambda z.z)(\lambda w.w)$.

$$\begin{aligned} & (\lambda x.xx)(\lambda y.((\lambda z.z)(\lambda w.w)y)) \\ \rightarrow & (\lambda y.((\lambda z.z)(\lambda w.w)y))(\lambda y.((\lambda z.z)(\lambda w.w)y)) \\ \rightarrow & \underline{(\lambda z.z)(\lambda w.w)}(\lambda y.(\underline{(\lambda z.z)(\lambda w.w)})y) \end{aligned}$$

Il paraît évident que les deux occurrences de $(\lambda z.z)(\lambda w.w)$ dans le terme final sont bien deux copies d'un même radical. Nous disons qu'elles sont deux *résidus* du radical $(\lambda z.z)(\lambda w.w)$ qui se trouvait dans le premier terme. Réduisons ces deux occurrences. Nous obtenons le terme suivant, dans lequel apparaissent deux nouveaux radicaux qui n'ont pas d'ancêtres visibles, et a fortiori pas d'ancêtre commun visible.

$$\underline{(\lambda w.w)(\lambda y.(\lambda w.w)y)}$$

Et pourtant, cet ancêtre existe bien quelque part. La figure 4 nous montre l'espace de réduction à partir du terme de départ $(\lambda x.xx)(\lambda y.((\lambda z.z)(\lambda w.w)y))$. Les deux radicaux qui nous intéressent correspondent aux deux pas de réduction indiqués en gras et annotés par w dans la partie la plus haute de la figure. Nous pouvons dans la figure remonter à un ancêtre commun, qui est le pas de réduction

$$(\lambda x.xx)(\lambda y.(\lambda w.w)y) \rightarrow (\lambda x.xx)(\lambda y.y)$$

également en gras et annoté par w dans la partie la plus basse de la figure. Ceci légitime une étape de calcul

$$(\lambda w.w)(\lambda y.(\lambda w.w)y) \Rightarrow (\lambda y.y)$$

qui réduirait les deux radicaux du terme $(\lambda w.w)(\lambda y.(\lambda w.w)y)$, mais pour voir ceci depuis notre séquence d'origine il faut en *permuter* les pas de calcul. Pour faciliter l'orientation, les pas de calcul de la figure sont annotés par le nom de la variable substituée.

Voilà une des manières dont Lévy a caractérisé les *familles* de radicaux émanant d'un *ancêtre* commun [Lé78, Lé80] : en considérant deux radicaux r_1 et r_2 au terme d'une séquence de réduction, s'il existe une permutation de la séquence dans laquelle les deux radicaux r_1 et r_2 sont des *résidus* d'un même

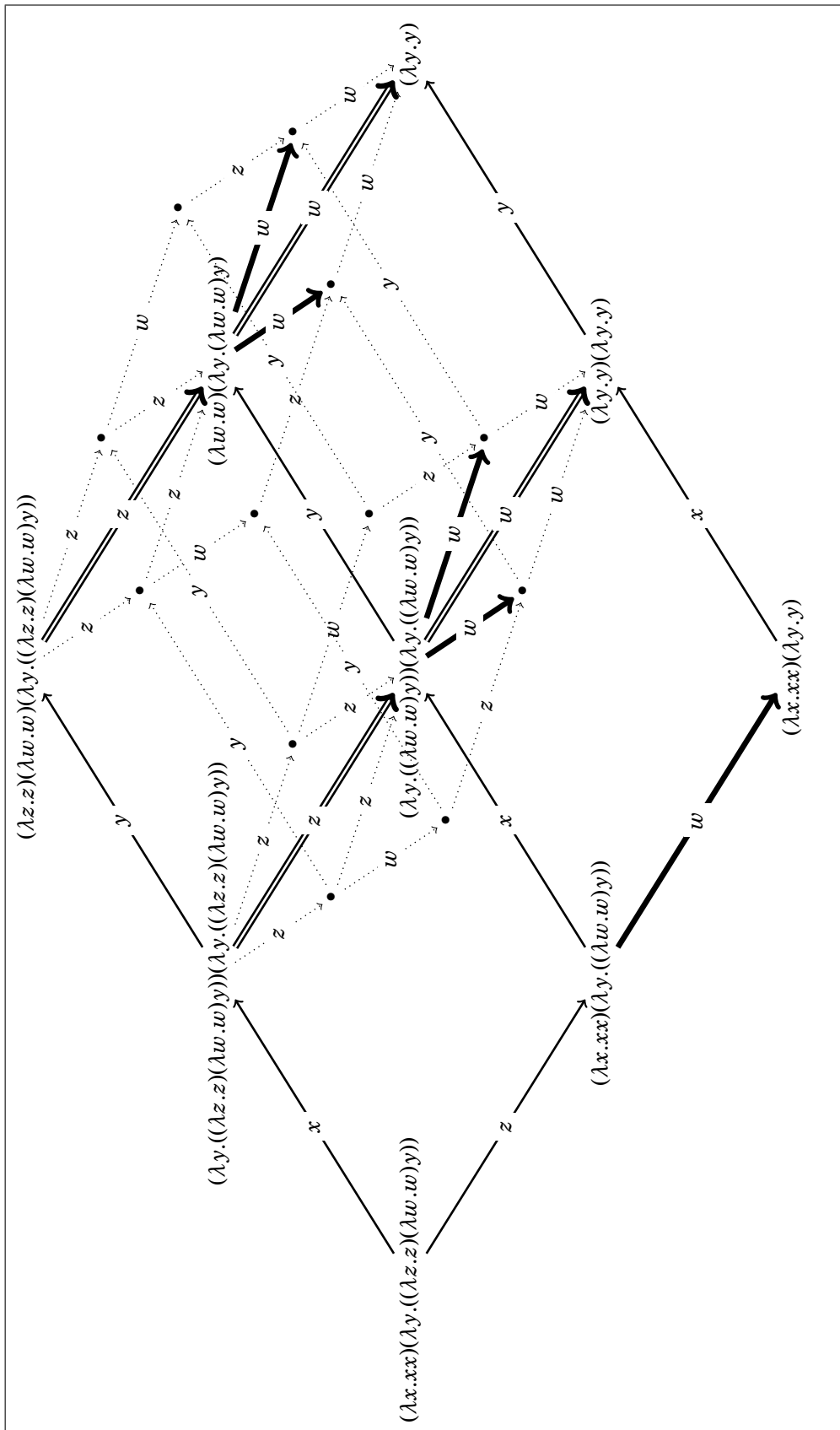


FIGURE 4: Ancêtre caché

radical r_0 (ce qui signifie que r_1 et r_2 *auraient pu* être réduits en un seul pas), alors les deux radicaux r_1 et r_2 sont de la même famille.

Il est possible ensuite de construire un espace de réduction dans lequel peut être réduit en une seule étape n'importe quel ensemble de radicaux appartenant à une même famille. La figure 4 montre un tel espace. Le souci est que cette construction demande de tenir compte de toutes les permutations de chaque séquence de calcul, ce qui n'est pas rien.

Lévy a contourné ce problème en apportant d'autres caractérisations des familles de radicaux. Une analyse est la suivante. Ce qui caractérise une famille est l'ensemble de ses ancêtres « les plus anciens ». La particularité d'un tel ancêtre premier est de ne pas être un résidu d'un radical antérieur mais d'être fraîchement *créé*, et les conditions de cette création sont un moyen d'identifier à la fois l'ancêtre et toute sa famille. Une famille peut donc être caractérisée par l'ensemble des *causes* de la création de ses plus vénérables ancêtres, où une « cause » désigne un événement *nécessairement* antérieur qui se retrouve dans toute permutation d'une séquence de calcul aboutissant à l'émergence de la famille observée.

À la figure 4 les pas de réduction annotés par x , y , z et w définissent quatre familles. Les familles x et z existent dès l'origine, en revanche aucun pas de calcul y ne peut apparaître avant qu'au moins un pas de calcul x ait été effectué : il y a une dépendance causale entre x et y (de même entre z et w).

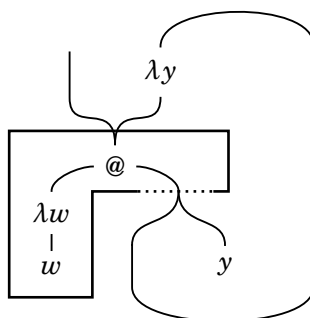
Pour rendre apparentes les familles de radicaux dans n'importe quelle séquence de calcul sans qu'il soit nécessaire d'en considérer les permutations, Lévy a proposé un système d'*étiquetage* pour le λ -calcul qui annote les termes avec l'ensemble des informations causales utiles.

Ainsi le travail de Lévy a permis de caractériser un espace de réduction dans lequel tous les pas de calcul légitimement partageables peuvent être partagés. Cet espace de réduction permet d'obtenir une borne inférieure sur la longueur des séquences de calcul évitant autant que nécessaire les duplications.

Cependant, à ce stade rien n'assurait que cette borne fût atteignable. Ce n'est d'ailleurs pas moins de dix ans plus tard qu'ont paru les deux premiers algorithmes réalisant la réduction optimale caractérisée par Lévy. L'un proposé par Kathail [Kat90] est resté méconnu, peut-être à cause de sa grande complexité. L'autre proposé par Lamping [Lam90] a été plus largement diffusé et a marqué une date, mais sans s'affranchir de la grande complexité qui semble intrinsèque à ce problème.

Pour permettre dans le cas du terme précédent $(\lambda w.w)(\lambda y.(\lambda w.w)y)$ que les deux radicaux soulignés soient partagés, les graphes de Wadsworth ne peuvent plus suffire. En effet les deux radicaux sont des termes différents, et ne peuvent donc pas être partagés en tant que « sous-termes égaux » comme cela se passe avec la technique de Wadsworth. Il faut maintenant un partage de fragments de termes, dans lequel en particulier l'expression $\lambda w.w$ peut être partagée, mais doit être appliquée à un « trou » pouvant contenir au choix la variable y ou l'expression $\lambda y.(\lambda w.w)y$. De plus, l'emboîtement des deux radicaux se traduit alors par l'apparition d'un cycle dans le graphe qui en rend la structure

significativement plus compliquée, comme dessiné informellement ci-dessous.



Cette difficulté technique de mise en œuvre des réduction optimales a été doublée de résultats négatifs sur leur *complexité*. En 1998, Asperti et Mairson ont montré [AM01] que le nombre d'étapes de réduction de familles de radicaux ne donnait pas d'indication fiable du coût réel d'exécution des programmes, car certaines de ces étapes de réduction peuvent nécessiter un temps d'exécution très grand.

Il est important de noter une chose à propos de ce résultat : il ne signifie pas que la réduction optimale puisse être moins bonne que les autres stratégies. Il signifie seulement qu'il convient de trouver d'autres mesures de complexité pour comparer dans le monde abstrait la réduction optimale à d'autres modèles d'évaluation. Mais historiquement, ce résultat semble avoir coûté cher à la réputation des réductions optimales, puisqu'il est parfois interprété comme l'impossibilité de donner une réalisation efficace à la réduction optimale à la Lévy.

Projet

La réaction à l'histoire tortueuse de l'optimalité et le constat de la réussite de solutions plus modestes sont parmi les motivations de mon projet, qui émane de la question suivante à propos du partage et des réductions efficaces :

Jusqu'où est-il possible d'aller avec des techniques d'évaluation simples ?

Mon travail peut donc être vu comme une expérience « d'optimalité sous contraintes ». L'objectif général est de trouver des modèles abstraits les meilleurs possibles pour l'évaluation efficace des programmes fonctionnels qui restent à la fois simples à comprendre intuitivement, simples à manipuler formellement, et simples à mettre en œuvre.

Cadre

Nous allons explorer les effets simplificateurs de deux restrictions de natures *a priori* différentes, l'une concernant le partage et l'autre les espaces de réduction.

Dans cette thèse nous allons explorer des stratégies de réduction avec partage dans lesquelles les entités partagées sont des sous-termes. Ce n'est pas le cas du partage optimal du λ -calcul, dans lequel nous avons vu que des contextes sont partagés. La technique d'évaluation de Wadsworth en revanche satisfait cette restriction, et nous espérons ainsi en hériter à la fois la manipulation intuitive de graphes à la structure plutôt simple et la possibilité d'une mise en œuvre assez naturelle à l'aide de techniques de mémoïsation [HJ76, FW76].

La deuxième restriction, qui concerne la dynamique des systèmes, est couramment appelée *réduction faible*. Si cette notion a été d'abord motivée par la comparaison entre le λ -calcul et la logique combinatoire [How70, Hin77, cH98], son intérêt pour nous vient d'une observation de ce qui est commun aux langages de programmation usuels. Considérons une expression $(\lambda x.t)a$ en λ -calcul, et rappelons les effets des stratégies usuelles :

- L'appel par valeur calcule d'abord le résultat de a , puis applique la fonction.
- L'appel par nom/nécessité applique directement la fonction (et met en place un mécanisme de partage).

Dans certains langages, ces deux disciplines peuvent également être mélangées pour traiter de manière particulière les fonctions et les structures de données qui le demandent. En revanche, en aucun cas l'expression t formant le corps de la fonction n'est évaluée avant l'instanciation de la fonction par application à son argument³. Cela signifie que le λ -calcul a une dynamique plus riche que les langages qu'il modélise, et que nous pouvons enlever de notre espace de réduction tout ou partie des termes et des pas de calcul qui concernent l'évaluation précoce du corps des fonctions sans altérer la capacité de notre système à modéliser les langages de programmation fonctionnelle. Nous allons donc appauvrir la dynamique du calcul pour passer en *réduction faible*, c'est-à-dire dans un mode où le calcul à l'intérieur d'une fonction est essentiellement interdit.

Le principal résultat de ce travail sera une relation entre ces deux restrictions, à savoir que le partage de sous-termes est suffisant pour réaliser le partage optimal en réduction faible, auquel nous associerons l'expression « *optimalité faible* ».

En effet, retirer des choses aux espaces de réduction a un impact sur les réductions optimales. D'une part, il est possible que certains pas de calcul supprimés fassent partie des chemins qui étaient les chemins les plus courts dans l'espace complet, ce qui nous force à utiliser d'autres chemins potentiellement plus longs. C'est ainsi que dans l'espace de réduction montré à la figure 5, dans lequel les pas de réduction interdits par la réduction faible sont omis ou indiqués en pointillé, il faut cinq pas pour atteindre le résultat en réduction faible alors que quatre étaient suffisants en réduction non-faible. D'autre part, la notion

3. Nous ne parlons ici que de ce qui se passe au moment de l'exécution, et pas des réductions couramment effectuées au moment de la compilation. Les optimisations apportées par les compilateurs apportent cependant un point qui comptera pour nous : certaines expressions constantes peuvent être extraites des fonctions, et donc être évaluées avant l'instanciation de la fonction dans laquelle elles apparaissent dans le programme source.

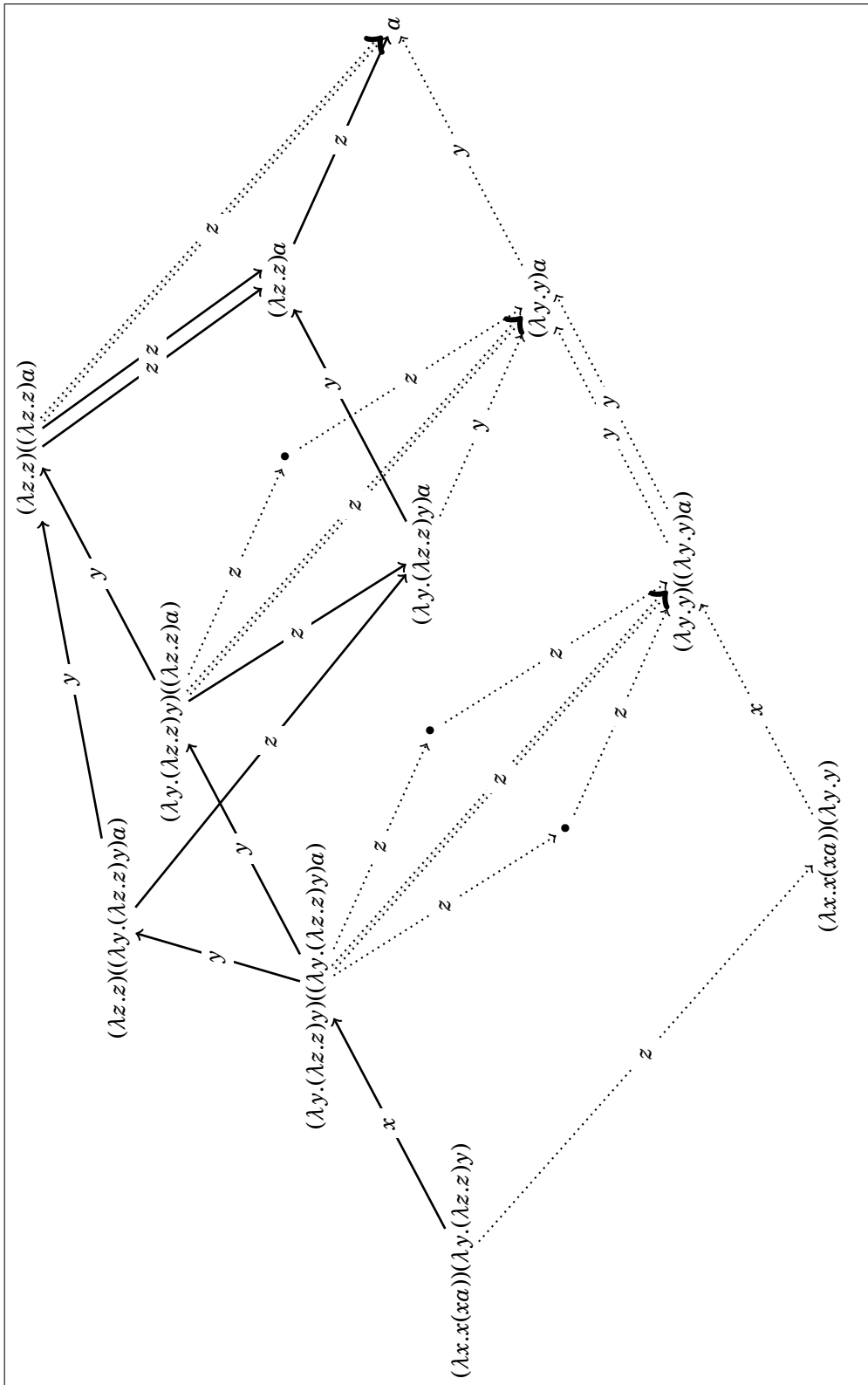


FIGURE 5: Espace de réduction faible

de famille de Lévy est elle-même modifiée. Cette dernière étant basée sur les possibles permutations des pas de calculs et les ancêtres communs que peuvent révéler ces permutations, la suppression de certains éléments de l'espace de réduction peut à la fois limiter les possibilités de permutation et faire disparaître certains ancêtres communs qui donnaient leur unité à une famille. Ainsi à la figure 5, alors que les pas de réduction associés aux λ -abstractions λz sont tous de la même famille en réduction non-faible de par leur ancêtre commun associé à $(\lambda z.z)y$ dans le terme de départ $(\lambda x.x(xa))(\lambda y.(\lambda z.z)y)$, ils sont séparés dans l'espace des réduction faibles.

Le λ -calcul, en tant que formalisme d'une concision extrême et bénéficiant d'une importante littérature, sera un point de départ. Cependant, notre cadre principal sera un formalisme de *réécriture d'ordre supérieur* [Acz78, Klo80, Nip91], dont les termes peuvent contenir plus de structure que les termes simples du λ -calcul, et qui permet d'effectuer des actions de réécriture combinant la manipulation d'objets structurés avec la manipulation de fonctions. La fonction `fold`, vue au début de cette introduction, qui applique récursivement une fonction à une liste de données est un exemple d'un tel système d'ordre supérieur. Plus précisément, nous utiliserons un formalisme dit du *second ordre*, à savoir le formalisme des *Combinatory Reduction Systems* (ou *CRS*) proposé par Klop [Klo80, KvOvR93], qui est suffisamment expressif pour la modélisation des langages de programmation fonctionnelle sans nous obliger à composer avec certaines difficultés techniques et conceptuelles de l'ordre supérieur général.

L'utilisation d'un tel formalisme dont le λ -calcul n'est qu'un cas particulier a une motivation évidente : gagner en *généralité* et augmenter ainsi la portée de nos résultats. Mais il y a deux autres avantages développés ci-dessous.

D'abord, un formalisme du second ordre offre un bon équilibre en ce qui concerne l'*adéquation* entre ce que nous observons dans le modèle abstrait (les longueurs des séquences de réduction dans les *CRS*) et la chose du monde réel sur laquelle nous voulons acquérir des informations (le temps d'exécution de programmes fonctionnels). Les *CRS* se placent en effet entre le λ -calcul, dans lequel l'encodage de structures de données se fait au prix de l'ajout d'un nombre potentiellement grand de pas de calcul risquant d'entraîner une surestimation du temps d'exécution, et l'ordre supérieur général, dans lequel l'application d'une règle de réduction cache une opération de filtrage potentiellement complexe et risque cette fois de nous conduire à une sous-estimation du temps d'exécution réel.

Ensuite, les systèmes du second ordre offrent encore un bon compromis en ce qui concerne la *simplicité* des manipulations formelles. Premièrement, les *CRS* sont naturellement plus faciles à manipuler que des formalismes plus généraux comme les *HRS* [Nip91]. Deuxièmement l'utilisation des *CRS*, bien que demandant un investissement de départ plus important que le λ -calcul, simplifie un aspect important du travail : grâce à son uniformité et son extensibilité naturelle, ce formalisme nous dispense de définir des extensions *ad hoc* à chaque fois que nous voulons ajouter des éléments à un système, comme il faudra le

faire sans cesse dans cette thèse dans les chapitres concernant le λ -calcul. Ce dernier point sera abondamment illustré au cours de cette thèse.

Ces travaux seront donc centrés sur le partage de sous-termes dans les *CRS* faibles. Enfin, ajoutons que pour les résultats concernant l’optimalité nous nous limiterons encore aux *CRS* dits *orthogonaux*, dans lesquels deux pas de réduction de même *source* ne peuvent interférer. Cette limitation est notamment ce qui permet de parler librement de permutation de pas de réduction et donc d’aborder la notion de causalité aussi naturellement qu’en λ -calcul.

Contributions

Je tiens avant tout à revendiquer un point : aucun des acteurs majeurs de cette histoire n’est un nouveau-venu tiré du néant par mes soins. Au contraire, les principaux concepts rencontrés et les principales techniques utilisées sont en filiation directe d’objets bien connus et ayant déjà eu pendant vingt à quarante ans un impact certain sur les scènes de la programmation fonctionnelle ou de la théorie de la réécriture. Cela concerne bien sûr les protagonistes déjà cités que sont l’évaluation pleinement paresseuse et la réduction optimale, mais également la plupart de ceux qui nous viendront en renfort par la suite.

En revanche, je me suis efforcé d’apprivoiser⁴ ces monstres sacrés, et de tailler à certains des habits neufs. Je revendique les avoir généralisés, autant que les avoir décalés et utilisés hors de leur contexte d’origine. Surtout, je revendique les avoir rassemblés dans une même histoire et les avoir fait communiquer. C’est dans ces nouveaux rapports entre des objets connus que je défends la principale originalité de ce travail.

La *thèse* qui structure tout ce document et en constitue le fil rouge est la suivante :

*L’évaluation pleinement paresseuse à la Wadsworth
est une réalisation de la réduction optimale à la Lévy
pour les systèmes de réécriture du second ordre orthogonaux faibles.*

Cet énoncé suppose trois contributions :

1. la généralisation de la notion de partage pleinement paresseux à la réécriture du second ordre,
2. l’application aux systèmes de réécriture du second ordre orthogonaux faibles de la théorie de l’optimalité à la Lévy, et
3. l’établissement du lien entre les deux notions obtenues,

auxquelles s’ajoutent trois contributions opportunistes :

4. la démonstration d’un lien, pour les systèmes de réécriture du second ordre orthogonaux faibles, entre les réductions optimales sans partage et les réductions optimales à la Lévy,

4. « C’est une chose trop oubliée, dit le renard. Cela signifie « créer des liens... ». » [dSE43]

5. la définition d'une mesure de coût pour la réduction avec partage pleinement paresseux, et
6. l'exploration de stratégies de réduction et de mécanismes de partage allant au-delà de l'optimalité faible.

Contribution 1. Partage pleinement paresseux.

La première contribution est la généralisation au cadre des *CRS* [Klo80, KvOvR93] de la technique de réduction de graphes de Wadsworth [Wad71]. Elle se décompose en deux parties :

- 1.a) définir un formalisme apte à exprimer la réduction partagée dans les *CRS*, et
- 1.b) décrire dans ce formalisme une réduction partagée qui correspond pas-à-pas avec celle de Wadsworth dans le cas particulier du λ -calcul.

Contribution 1.a) Un formalisme pour la réduction partagée.

Des graphes similaires aux λ -graphes de Wadsworth ont déjà été proposés à la fois pour la réécriture du premier ordre [BvEG⁺87, AK96, Plu99, Blo01] et la réécriture d'ordre supérieur [Kah98, Has97, BBCK07]. Cependant ces formalismes sont délicats à manipuler : d'une part l'identification du membre gauche d'une règle dans un graphe donné requiert une notion de morphisme de graphes, et d'autre part les graphes représentant des termes d'ordre supérieur souffrent de complications dues aux phénomènes de liaison de variables.

Pour contourner cette difficulté de formalisation nous utiliserons une représentation alternative inspirée des travaux de Maranget [Mar92, Mar91] dans laquelle les graphes sont représentés par des termes étiquetés et les réductions de graphes par des séquences de réduction sur ces termes étiquetés. Le principe de cette représentation est illustré ci-dessous, où les deux occurrences de a^δ (soulignées) portent une même étiquette, ce qui signifie qu'elles représentent le même nœud de graphe (celui marqué d'une étoile) et sont à réduire en une seule étape.

$$\Rightarrow \begin{array}{l} f^\alpha(f^\beta(a^\gamma, \underline{a^\delta}), \underline{a^\delta}) \\ f^\alpha(f^\beta(a^\gamma, \underline{b^\zeta}), \underline{b^\zeta}) \end{array} \quad \Bigg| \quad \begin{array}{c} \begin{array}{c} \curvearrowright f \\ a \quad \quad \quad a^* \end{array} \\ \Rightarrow \begin{array}{c} \curvearrowright f \\ a \quad \quad \quad b \end{array} \end{array}$$

Cette manière de formaliser la réécriture de graphes a déjà été utilisée pour les systèmes de réécriture du premier ordre orthogonaux [Mar92] ainsi que pour le λ -calcul faible [BLM05, BLM07]. Elle sera ici généralisée d'une manière qui permet notamment d'atteindre tous les *CRS*, faibles ou non, orthogonaux ou non.

Pour cela, nous introduirons à la partie I les *systèmes de partage étiquetés* (*Sharing-via-Labeling Systems, SLS*, [Bal12a]), un formalisme abstrait manipulant des termes étiquetés et muni d'axiomes qui :

- garantissent que les étiquettes d'un terme t décrivent la structure d'un graphe \mathcal{G}_t réalisant un partage de certains sous-termes de t , et

- permettent de définir les séquences de réduction de termes étiquetés à partir de t qui simulent les pas de réduction de graphe à partir de \mathcal{G}_t .

Ce cadre abstrait ne fait aucune hypothèse sur la forme des règles de réécriture des systèmes concrets, et s'applique donc aussi bien aux formalismes étiquetés déjà existants [Mar92, BLM07] qu'à de nouveaux étiquetages sur des systèmes potentiellement plus compliqués. Le formalisme axiomatique sera justement utilisé dans chacun des dix chapitres des parties II à V sur des systèmes variés comprenant : des systèmes du premier ordre orthogonaux, le λ -calcul faible, des *CRS* faibles et orthogonaux, des *CRS* non-faibles et non-orthogonaux.

Contribution 1.b) Expression du partage pleinement paresseux.

Wadsworth a été le premier à décrire le partage pleinement paresseux pour le λ -calcul [Wad71]. La notion a ensuite été reprise plusieurs fois, toujours à partir du λ -calcul, mais avec six présentations très différentes [Tur79, PJ87, AF97, Ses97, SW10, BLM07].

En utilisant le formalisme axiomatique défini à la partie I nous proposerons à la partie II une notion de réduction avec partage pleinement paresseux pour les *CRS* faibles qui généralise la notion existante en λ -calcul. Cette contribution comporte deux parties : d'abord une analyse du partage pleinement paresseux en λ -calcul puis son extension aux *CRS* faibles.

Nous exprimerons d'abord la technique de réduction de λ -graphes de Wadsworth (chapitre 2) et quatre des six autres descriptions du partage pleinement paresseux [AF97, Ses97, SW10, BLM07] (chapitre 3) dans le cadre axiomatique. Cette représentation sera faite dans un unique λ -calcul faible étiqueté muni d'un paramètre permettant d'ajuster le comportement des étiquettes à chacun des cinq systèmes (la technique originale de Wadsworth et les quatre autres concernés [AF97, Ses97, SW10, BLM07]). Point notable : il sera apparent dans cette représentation homogène que les cinq systèmes ne définissent *pas* le même partage de sous-termes.

Cependant, et ce sera le résultat principal du chapitre 3, nous montrerons que ces cinq systèmes sont bisimilaires [Bal12b]. Autrement dit, ils définissent tous la même notion de réduction partagée, et les divergences qu'ils peuvent afficher sur le partage de certains sous-termes ne concernent jamais des radicaux. Nous en déduirons qu'il n'existe donc bien qu'une notion de réduction avec partage pleinement paresseux dans la littérature du λ -calcul⁵.

Nous proposerons ensuite un étiquetage pour les *CRS* faibles (chapitre 4) en s'inspirant du λ -calcul étiqueté représentant la technique de Wadsworth. Cet étiquetage aura pour caractéristique d'être fait à l'intérieur même du formalisme des *CRS* auquel il s'applique. Nous montrerons que sur le cas particulier du λ -calcul faible cet étiquetage des *CRS* faibles spécifie une réduction partagée bisimilaire à celle de Wadsworth, et donc également bisimilaire aux autres présentations du partage pleinement paresseux pour le λ -calcul. Ainsi l'étiquette-

5. Une sixième description [PJ87] sera encore unifiée avec les autres à la partie IV. La septième [Tur79] ne sera pas traitée dans cette thèse car elle utilise une traduction vers un système de combinateurs qui ne préserve pas le nombre de pas de réduction.

tage proposé pour les *CRS* faibles sera une généralisation aux *CRS* légitime du partage pleinement paresseux tel qu'il existe à ce jour pour le λ -calcul.

Contribution 2. Optimalité pour les *CRS* faibles orthogonaux.

La deuxième contribution est une caractérisation des réductions optimales dans les *CRS* faibles orthogonaux. Elle est basée sur une description axiomatique proposée par Glauert et Khasidashvili [GK96], qui interprète les travaux de Lévy [Lé78, Lé80] dans un cadre abstrait en mettant l'accent sur la notion de dépendance causale. Cette contribution se décompose en trois parties :

- 2.a) analyser la structure causale des *CRS* faibles orthogonaux,
- 2.b) les réduire à des systèmes de réécriture du premier ordre orthogonaux, qui sont plus simples, et
- 2.c) instancier sur ces derniers le cadre abstrait de Glauert et Khasidashvili.

Contribution 2.a) Analyse de la causalité.

La structure causale des réductions du λ -calcul est déjà bien connue grâce à la théorie des résidus développée par Lévy [Lé78] et à la description des permutations de pas de réduction associée. Cette analyse a également été portée au cadre des systèmes de réécriture du premier ordre orthogonaux [Mar92, Ter03] et esquissée pour le λ -calcul faible [BLM07] avec la définition d'un système étiqueté.

Nous décrivons les dépendances causales entre les pas de réduction d'un *CRS* faible orthogonal à la partie III au moyen d'une méthode d'étiquetage des termes. La relation de causalité décrite par les étiquettes sera prouvée correcte et complète vis-à-vis d'un critère abstrait fondé sur les permutations de pas de réduction et correspondant en particulier au critère utilisé par Glauert et Khasidashvili [GK96]. Cette analyse sera étendue progressivement du λ -calcul faible aux *CRS* faibles orthogonaux.

Nous partirons au chapitre 5 du λ -calcul faible étiqueté proposé par Blanc, Lévy et Maranget [BLM07] et détaillerons comment les étiquettes de ce calcul, en plus de décrire une réduction partagée (ce qu'avaient déjà montré ses auteurs), décrivent précisément les relations de dépendance entre les différents pas de réduction.

Nous analyserons ensuite au chapitre 6 le cas plus compliqué du *Pure Pattern Calculus* de Jay et Kesner [JK06, JK08, Jay09, JK09], pour lequel nous définirons un système d'étiquettes plus évolué capable de traiter les règles de ce calcul basé sur une méta-théorie riche [Bal10a]. Nous verrons sur cet exemple comment la non-séquentialité d'un système rend plus complexes les relations de dépendance causale entre ses pas de réduction, et interfère avec la représentation de ces dernières dans des étiquettes.

Enfin, au chapitre 7 nous définirons un nouvel étiquetage des *CRS* faibles en décrivant la structure causale, et nous prouverons que dans le cas de *CRS* faibles orthogonaux la relation de causalité décrite par les étiquettes est correcte et complète vis-à-vis d'un critère abstrait fondé sur les permutations de pas de réduction [GK96].

Contribution 2.b) Réduction au premier ordre.

La mise en œuvre des langages fonctionnels fait naturellement appel à des transformations de programmes qui, peu ou prou, contiennent une réduction des fonctions d'ordre supérieur à des structures plus primitives. Ces transformations incluent par exemple la défonctionnalisation ou le λ -*lifting* [Rey98, Joh85, DN01].

Ces opérations usuelles sous-entendent une adéquation de la réécriture du premier ordre avec le λ -calcul faible, qui a été confirmée par certains outils plus théoriques comme les traductions en logique combinatoire [CF74, How70, Hin77, Tur79] ou les substitutions explicites [ACCL90, Kes07]. Cependant, et c'est notamment visible avec les substitutions explicites, ces transformations sont susceptibles de rompre l'orthogonalité du système d'origine et d'en décomposer les pas de réduction d'une manière qui rend le lien avec le premier ordre quelque peu distendu.

Nous formaliserons à la partie IV la technique bien connue de λ -*lifting* [Joh85, PJ87] en tant que transformation d'un CRS faible en un système de réécriture du premier ordre, transformation dont nous montrerons qu'elle définit entre sa source et sa cible une bisimulation qui préserve toutes les propriétés manipulées dans cette thèse. C'est-à-dire que cette transformation préserve non seulement les séquences de réduction et leurs longueurs, mais aussi l'éventuelle orthogonalité des systèmes, leur structure causale, et le partage.

Cette formalisation sera menée d'abord dans le cadre du λ -calcul (chapitre 8) comme une interprétation directe du λ -*lifting* pleinement paresseux déjà existant [Hug82, PJ87], puis généralisée aux CRS faibles (orthogonaux ou non). Dans les deux cas, mais ce sera plus particulièrement élégant avec les CRS qui s'y prêtent naturellement, la transformation de *lifting* sera elle-même décrite comme un système de réécriture d'ordre supérieur. Ceci permettra de considérer les étapes intermédiaires de la transformation, et donc de déduire ses bonnes propriétés par des raisonnements purement locaux.

Contribution 2.c) Application des théorèmes généraux.

La partie V (chapitre 10) combine les constructions des deux parties précédentes avec le cadre abstrait de Glauert et Khasidashvili [GK96] pour caractériser la réduction optimale à la Lévy dans les CRS faibles orthogonaux.

Nous verrons notamment que l'étiquetage des CRS faibles orthogonaux défini à la partie III, car il reflète fidèlement leur structure causale, permet de définir des *familles faibles* de radicaux vérifiant la même propriété d'optimalité que les familles de Lévy sur le λ -calcul non-faible [Lé78, Lé80] : toute séquence réduisant à chaque étape une famille complète dont l'un des éléments au moins est nécessaire est de longueur minimale dans l'espace de réduction dans lequel toute partie d'une famille peut être réduite en une étape.

L'étiquetage causal défini à la partie III permettra de définir les familles, tandis que la réduction au premier ordre mise en place à la partie IV aura pour rôle de simplifier la vérification des axiomes.

Contribution 3. Équivalence.

La troisième contribution est transverse et consiste à assurer au fur et à mesure du développement des parties III, IV, et V la correspondance des nouvelles constructions avec la définition de la pleine paresse posée à la partie II.

L'existence d'un lien entre la réduction pleinement paresseuse de Wadsworth pour le λ -calcul et la structure causale du λ -calcul faible a été suggérée par Blanc, Lévy et Maranget [BLM07]. Cette thèse formalise ce lien dans le cadre plus général des CRS faibles orthogonaux, en démontrant l'équivalence entre le partage pleinement paresseux défini à la partie II et les familles faibles de radicaux décrites par l'étiquetage causal proposé à la partie III.

À ce stade, la démonstration de cette équivalence est quasiment immédiate grâce aux formalisations très proches qui ont été données aux deux notions : le partage pleinement paresseux comme la causalité sont décrits par deux étiquetages d'un même système de réécriture sous-jacent. Il suffit donc de prouver que ces deux étiquetages spécifient le même partage, ce qui est fait à la partie III dès la définition du deuxième de ces systèmes. De cette équivalence nous déduisons que le résultat d'optimalité sur la réduction des familles faibles complètes nécessaires de la partie V s'applique également à la réduction pleinement paresseuse.

Contribution 4. Le partage au secours de la calculabilité.

Notre progression vers la démonstration de la thèse a comme sous-produit intéressant un lien entre la réduction pleinement paresseuse et la réduction optimale dans un espace de réduction faible et sans partage. Fait amusant : la première est souvent calculable, alors que la seconde ne l'est pas. Ce nouveau tableau se découpe en trois parties :

- 4.a) démonstration de la non-calculabilité des stratégies de réduction optimales sans partage pour le λ -calcul faible,
- 4.b) caractérisation de stratégies (nécessairement non-calculables) optimales pour les CRS faibles orthogonaux sans partage : la réduction nécessaire profonde (*innermost needed reduction*), puis
- 4.c) démonstration de l'équivalence (en termes de longueurs de séquences de réduction) entre les réductions nécessaires profondes et les réductions pleinement paresseuses.

Contribution 4.a) Non-calculabilité.

Barendregt, Bergstra, Klop et Volken ont prouvé qu'une stratégie de réduction sans partage optimale pour le λ -calcul ne pouvait être calculable [BBKV76, Bar84]. Une étape de leur preuve repose sur la réduction non-faible du λ -calcul et empêche donc le transfert immédiat au λ -calcul faible.

Nous réalisons le transfert de ce résultat du λ -calcul non-faible au λ -calcul faible à la partie V et prouvons donc qu'une stratégie de réduction sans partage optimale pour le λ -calcul faible ne peut être calculable. Pour ceci nous utilisons la décomposition des λ -termes en supercombinateurs suggérée par la transformation de λ -*lifting* pleinement paresseux [Hug82, PJ87].

Contribution 4.b) Optimalité des réductions nécessaires profondes.

Nous verrons que les stratégies de réduction pour les *CRS* faibles orthogonaux qui effectuent toujours un pas de réduction le plus profond parmi les pas de réduction nécessaires sont optimales, dans le sens qu'aucune séquence de réduction sans partage entre les mêmes termes ne peut être plus courte. Ce résultat est obtenu à partir d'un résultat de Khasidashvili [Kha93] énonçant le même critère d'optimalité pour les systèmes de réécriture du premier ordre orthogonaux. Le transfert de ce critère vers l'ordre supérieur est effectué grâce à la réduction des *CRS* faibles au premier ordre préservant la nécessité et l'orthogonalité introduite à la partie IV.

À travers les *CRS* faibles nous démontrons donc que ce critère d'optimalité vaut encore pour le λ -calcul faible. Or Guerrini a exhibé un contre-exemple [Gue96] montrant que ce critère ne donnait pas des séquences de réduction optimales dans le λ -calcul non faible. Cette divergence est un nouvel indice du fait que le λ -calcul faible, bien qu'ayant la forme du λ -calcul, est plus proche par ses propriétés de la réécriture du premier ordre.

Contribution 4.c) Pleine paresse sans partage.

Un rapprochement entre le fait que la réduction partagée dans les systèmes de réécriture du premier ordre vérifie la propriété du diamant et des remarques de Glauert et Khasidashvili sur les propriétés des radicaux nécessaires [GK96] suggère qu'il existe dans les systèmes de réécriture du premier ordre orthogonaux des séquences de réduction qui n'effectuent que des pas de réduction nécessaires et ne provoquent aucune duplication. Autrement dit ces séquences seraient aussi bonnes que les séquences optimales avec partage. Nous interprétons ceci à la partie V dans le cadre général des *CRS* faibles orthogonaux et prouvons que les réductions optimales sans partage ont même longueur que les réductions avec partage pleinement paresseux de radicaux nécessaires.

De cette équivalence nous déduisons que dans les *CRS* faibles orthogonaux la réduction optimale à la Lévy peut être atteinte sans faire appel au partage. Nous savions déjà qu'elle était équivalente à la réduction pleinement paresseuse, et nous ajoutons maintenant qu'elle est équivalente aux stratégies optimales sans partage dont une des caractéristiques est de ne pas pouvoir être calculables [BBKV76].

Remarquons d'abord que la non-calculabilité des stratégies optimales sans partage est liée au fait que la nécessité d'un pas de réduction n'est pas décidable en général [BKKS87], ce qui rend difficile la recherche d'un radical nécessaire *le plus profond*. Le partage pleinement paresseux, en neutralisant l'effet des duplications, fait qu'il n'est plus utile de commencer par le pas de réduction nécessaire le plus profond : toute étape de réduction nécessaire fait partie d'une séquence de longueur minimale. Ainsi, il suffit de savoir trouver *un* radical nécessaire, ce qui est simple dans des systèmes comme le λ -calcul.

Ainsi, dans ces systèmes passer de la réduction sans partage à la réduction avec partage permet d'obtenir une stratégie calculable *équivalente* à la stratégie

optimale non-calculable. Cette équivalence se manifeste par deux propriétés : la séquence de réduction avec partage pleinement paresseux est une *permutation* de la séquence optimale sans partage, et les deux ont *même longueur*.

Contribution 5. Complexité polynomiale.

Le λ -calcul a longtemps été coincé entre deux remarques de complexité : d'une part le nombre de pas d'une séquence de β -réduction sans partage n'est pas directement un moyen d'évaluer son coût d'exécution puisque la β -réduction sans partage implique des duplications de sous-termes dont la taille peut exploser, et d'autre part le nombre de pas de réduction avec partage optimal n'est pas non plus une bonne évaluation [AM01]. Cependant, Dal Lago et Martini ont montré que des λ -calculs en appel par valeur ou en appel paresseux, qui représentent un niveau intermédiaire entre réduction non partagée générale et partage optimal, avaient de meilleurs propriétés de complexité [LM09a].

À la partie II nous généralisons ces travaux à la réduction partagée pleinement paresseuse des *CRS* faibles (pas forcément orthogonaux). Nous définissons d'abord une mesure de complexité basée sur la croissance des graphes au cours d'une séquence de réduction. Puis, dans une classe de *CRS* faibles contenant le λ -calcul, nous montrons que le nombre d'étapes de réduction avec partage pleinement paresseux est polynomialement lié au nombre de transitions nécessaires pour effectuer le même calcul sur une machine de Turing. Pour cela, nous utilisons les étiquettes spécifiant le partage pour montrer que la taille des graphes d'une séquence de réduction avec partage pleinement paresseux n'explose pas. Ce résultat forme un premier pas dans la vérification de l'adéquation du modèle de la réduction partagée pleinement paresseuse avec ses mises en œuvre.

Contribution 6. Aller plus loin.

La pleine paresse réalise une *optimalité faible*, c'est-à-dire une optimalité relative à certaines contraintes de réduction faible qui permettent de préserver la simplicité du modèle. Cependant il est possible d'aller plus loin que cette optimalité faible tout en continuant à limiter le partage à des sous-termes et en restant dans le cadre simple des formalismes utilisés pour décrire et réaliser l'optimalité faible. La partie V (chapitre 11) développe deux moyens d'aller *au-delà* de l'optimalité faible :

- 6.a) combiner le partage pleinement paresseux avec de la réduction non-faible, et
- 6.b) optimiser le partage pleinement paresseux.

Contribution 6.a) Réduction non-faible.

La simple non-correspondance entre les notions d'optimalité à la Lévy pour les versions faibles ou non du λ -calcul est un indice fort de la capacité d'optimisation supérieure du λ -calcul non-faible. À titre d'exemple concret de cette optimisation la réduction dite *complètement paresseuse* [HG91, Amt93, Sin08], basée sur des mécanismes d'évaluation partielle, a été décrite par Holst et Gomard comme réalisant un partage *plus-que-pleinement* paresseux [HG91].

Ce qui est incertain est la possibilité de réaliser ces optimisations dans un cadre simple où le partage est limité à des sous-termes. En effet, s'il est de notoriété publique que le partage optimal nécessite de partager des contextes, nous ne savons pas à quel niveau de la hiérarchie paresseuse commence cette nécessité. En particulier, bien que Sinot ait montré que l'utilisation de contextes était pratique dans la définition d'une sémantique de la paresse complète [Sin08], leur nécessité ne semble pas établie.

Nous montrons au chapitre 11 que le partage pleinement paresseux peut être combiné avec une réduction non-faible qui permet, dans l'esprit de l'évaluation complètement paresseuse de Holst et Gomard, de combiner des réductions à l'intérieur des fonctions avec du partage de sous-termes pour aller au-delà de la pleine paresse. Ceci sera fait comme dans toutes les parties précédentes avec un système de partage de sous-termes spécifié par une notion de réduction étiquetée.

Contribution 6.b) Partage optimisé.

Comme cela est vérifié à la contribution 1.b), différentes manières d'interpréter le partage pleinement paresseux de Wadsworth sont équivalentes en réduction faible. Cependant, une fois la restriction à la réduction faible levée ces différentes variations sont à reconsidérer.

En particulier, l'approche par Blanc, Lévy et Maranget [BLM07] apporte un comportement de partage plus riche qui autorise le partage de deux sous-termes dont les variables sont liées par des lieux différents. Les graphes obtenus, qualifiés de *non-admissibles* par Wadsworth, sont tout à fait naturels vus sous l'angle des termes étiquetés, pour peu que la gestion des noms de variables soit adaptée à cette situation. Cette possibilité est discutée au chapitre 11 dans le cadre général des *CRS*, et une stratégie concrète est proposée pour une extension du λ -calcul.

Organisation du document et parcours de lecture

Ce document est organisé en cinq parties :

- I. Partage : définition du formalisme axiomatique de termes étiquetés utilisé dans tout le développement subséquent pour spécifier la réduction partagée.
- II. Pleine paresse : passage en revue de la littérature sur la pleine paresse en λ -calcul et généralisation au cadre d'ordre supérieur des *Combinatory Reduction Systems (CRS)*.
- III. Causalité : enrichissement des systèmes de la partie II pour y incorporer des informations sur la structure causale des espaces de réduction, qui participe au fondement de l'optimalité à la Lévy.
- IV. Premier ordre : établissement d'un lien fort entre les *CRS* faibles et des systèmes du premier ordre, qui préserve l'orthogonalité ainsi que les propriétés de partage et de causalité, et permet de simplifier les preuves des résultats d'optimalité.
- V. Optimalités : démonstration par l'utilisation des étiquettes causales et par réduction au premier ordre de l'optimalité de la réduction pleinement paresseuse dans les *CRS* faibles orthogonaux.

Le cœur du développement est mené conjointement sur le λ -calcul et sur les *CRS*. La table suivante résume le découpage de ces cinq parties en chapitres et précise les systèmes concernés.

I. Partage	II. Pleine paresse	III. Causalité	IV. Premier ordre	V. Optimalités
	2. λ -calcul	5. λ -calcul		
			8. λ -calcul	10. <i>CRS</i> (et λ)
1. Abstrait	3. Variations λ	6. <i>PPC</i>		
	4. <i>CRS</i>	7. <i>CRS</i>	9. <i>CRS</i>	11. <i>CRS</i> (et λ)

Ainsi les parties II, III, et IV commencent toutes trois par un chapitre traitant le λ -calcul et terminent par un chapitre généralisant aux *CRS*. La partie I développe un formalisme abstrait et n'est donc pas soumise à ce découpage, et la partie V se permet de ne traiter explicitement que le cadre plus général des *CRS* car la réduction au premier ordre gomme la différence entre λ -calcul et *CRS*.

Deux chapitres additionnels sont intercalés dans cette organisation :

- Le chapitre 3, au centre de la partie II, passe en revue les principales variations sur l'évaluation pleinement paresseuse existant dans la littérature et montre leur unité.
- Le chapitre 6, au centre de la partie III, montre sur le cas du *Pure Pattern Calculus* les difficultés qu'il peut y avoir à parler de causalité dans des cadres plus généraux que le λ -calcul.

Enfin, après le chapitre 10 énonçant les résultats d’optimalité, la partie V contient un ultime chapitre 11 relativisant ces résultats d’optimalité et montrant en particulier comment les outils de cette thèse permettent d’aller plus loin.

Outre la lecture intégrale, plusieurs parcours sont possibles et apportent des contributions cohérentes :

- Démonstration de l’unité de la notion d’évaluation pleinement paresseuse en λ -calcul telle qu’elle apparaît disséminée dans les littératures des mondes de la réécriture et de la programmation fonctionnelle : chapitres 1, 2, 3, 8. Ce parcours correspond à l’article [Bal12b].
- Démonstration de l’équivalence entre la technique d’évaluation de Wadsworth et la réduction optimale à la Lévy appliquée au λ -calcul faible : chapitres 1, 2, 5, 8, 10.
- Définition de la réduction pleinement paresseuse dans les *CRS* faibles et démonstration de son équivalence avec la réduction optimale à la Lévy pour les *CRS* faibles orthogonaux : chapitres 1, 4, 7, 9, 10. Ceci est le chemin le plus court vers l’établissement de la *thèse* supportée par ce document.
- Définition de stratégies de réduction pour le λ -calcul et les *CRS* qui vont au-delà de la pleine paresse tout en se contentant d’un formalisme avec partage de sous-termes : chapitres 1, 11.

Méthodes

Au moment de me lancer dans de tels travaux dans un domaine plutôt formel et technique, je formule un vœu : que le formalisme ne soit pas un frein.

C’est à ceci que tentent de répondre les deux aspects méthodologiques de mes travaux que je souhaite finalement mettre en avant : l’utilisation quand cela est possible d’une approche axiomatique qui se soucie plus des propriétés des systèmes que de leur forme, et la création de liens entre les formalismes qui permet un transfert de propriétés depuis les systèmes les plus simples ou les mieux étudiés vers les systèmes les plus neufs ou les plus complexes.

Ainsi, il pourra être remarqué que nombre des principales preuves contenues dans cette thèse sont soit des vérifications d’axiomes qui permettent ensuite d’hériter les propriétés voulues d’un cadre abstrait, soit des théorèmes de bi-simulation qui permettent de se réduire à des formalismes dans lesquels les propriétés désirées sont plus commodes à obtenir voire déjà connues.

L’approche axiomatique apparaît en particulier avec l’utilisation au chapitre culminant 10 de la description abstraite de Glauert et Khasidashvili des réductions optimales, et avec le développement au chapitre 1 et l’utilisation d’un bout à l’autre de la thèse des systèmes de partage étiquetés *SLS*.

La réduction à des formalismes plus simples est quant à elle particulièrement vivace avec l’utilisation des *SLS* qui permettent de troquer un formalisme de graphes contre un formalisme de termes, et avec la réduction au premier ordre mise en place aux chapitres 8 et 9, deux points qui nous permettent finalement de définir des systèmes de réduction de graphes pour la réécriture d’ordre supérieur,

sans graphes, ni ordre supérieur.

Matériel

Sommaire

0.1 Relations binaires	35
0.2 Réécriture abstraite	36
0.3 Signatures, termes, contextes	41
0.4 Réécriture du premier ordre	42
0.5 Lambda-calcul	44
0.6 <i>Combinatory Reduction Systems</i>	48

Ce chapitre préliminaire regroupe des définitions et notations utiles à l'ensemble de la thèse. Il mêle des rappels de la littérature à des conventions et ajouts spécifiques à cette thèse. La section 0.1 rappelle le vocabulaire général des relations binaires. La section 0.2 regroupe des définitions de réécriture abstraites issues de différentes sources. La section 0.3 exprime les notions usuelles relatives à la structure de terme dans un cadre général. Les sections 0.4, 0.5 et 0.6 enfin redonnent les définitions usuelles de la réécriture du premier ordre, du λ -calcul et du formalisme d'ordre supérieur des *Combinatory Reduction Systems* (CRS) et leur adjoignent quelques vocabulaires utiles à la suite.

0.1 Relations binaires

Une *relation binaire* sur un ensemble exprime des liens entre les différents éléments de cet ensemble. Les relations rencontrées dans cette thèse exprimeront notamment :

- L'existence d'un ordre permettant de comparer les éléments.
- La succession ou l'évolution temporelle des éléments.
- La ressemblance entre éléments ou leur appartenance à des groupes.

Définition 0.1 (Relation binaire). *Une relation binaire R sur un ensemble E est un sous-ensemble de $E \times E$. L'appartenance à R d'un couple $(x, y) \in E \times E$ est notée xRy .*

Définition 0.2 (Vocabulaire des relations binaires). *Soit R une relation binaire sur un ensemble E . La relation R est dite. . .*

réflexive, si pour tout $x \in E$, xRx .

transitive, si pour tous $x, y, z \in E$, si xRy et yRz alors xRz .

symétrique, si pour tous $x, y \in E$, si xRy alors yRx .

anti-symétrique, si pour tous $x, y \in E$, si xRy et yRx alors $x = y$.

Définition 0.3 (Clôture d'une relation binaire). La **clôture** d'une relation binaire R par rapport à l'une quelconque des quatre caractéristiques précédentes est la plus petite relation binaire contenant R et vérifiant ladite caractéristique.

Nous rencontrerons notamment ici la clôture transitive.

Définition 0.4 (Relations particulières). Les deux types de relations binaires suivants sont particulièrement utilisés.

Ordre : relation binaire réflexive, transitive, et anti-symétrique.

Équivalence : relation binaire réflexive, transitive, et symétrique.

0.2 Réécriture abstraite

La réécriture abstraite [GK96, Mel96, Ter03] est la description la plus générale de la réécriture. Elle n'observe que des transitions (appelées *pas de réduction*) entre objets, et ne fait aucune mention de la forme des objets ni des règles sous-tendant les transitions.

Un système abstrait de réécriture est composé d'un ensemble \mathcal{O} d'**objets** et d'un ensemble \mathcal{R} de **pas de réduction** où chaque pas de réduction est orienté d'un objet **source** vers un objet **cible**.

Définition 0.5 (ARS). Un **système abstrait de réécriture** (Abstract Rewriting System, ARS) est un quadruplet $(\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ où :

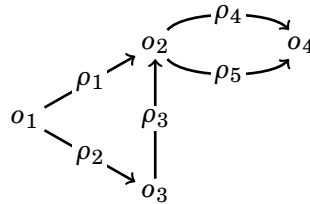
- \mathcal{O} et \mathcal{R} sont des ensembles dénombrables, et
- src et tgt sont deux fonctions totales de l'ensemble de pas de réduction \mathcal{R} dans l'ensemble d'objets \mathcal{O} .

Pour tout pas de réduction $\rho \in \mathcal{R}$, si $\text{src}(\rho) = o$ et $\text{tgt}(\rho) = o'$ alors nous notons encore $\rho : o \rightarrow o'$.

Un système abstrait de réécriture $(\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ peut être vu comme un multigraphe orienté dont les sommets sont les objets $o \in \mathcal{O}$ et les arcs les pas de réduction $\rho \in \mathcal{R}$.

Exemple 0.6 (Multigraphe).

Dans le système dessiné ci-dessous, nous avons $\text{src}(\rho_4) = \text{src}(\rho_5) = o_2$ et $\text{tgt}(\rho_4) = \text{tgt}(\rho_5) = o_4$. Les pas de réductions ρ_4 et ρ_5 ont donc même source et même cible, mais n'en sont pas moins distincts l'un de l'autre.



Définition 0.7 (Séquences de réduction). Soit $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ un ARS. Une **séquence de réduction** est une séquence $\rho = \rho_1 \dots \rho_n$ de pas de réduction telle que pour tout $2 \leq i \leq n$ nous avons $\text{src}(\rho_i) = \text{tgt}(\rho_{i-1})$. Les notions de **source** et **cible** s'étendent à une séquence de réduction $\rho = \rho_1 \dots \rho_n$ par $\text{src}(\rho) = \text{src}(\rho_1)$ et $\text{tgt}(\rho) = \text{tgt}(\rho_n)$. Pour toute séquence de réduction ρ , si $\text{src}(\rho) = o$ et $\text{tgt}(\rho) = o'$ alors la séquence de réduction ρ est encore notée $\rho : o \rightarrow o'$. Pour tout objet $o \in \mathcal{O}$ il existe une séquence de réduction vide $\epsilon_o : o \rightarrow o$.

Exemple 0.8.

Deux exemples de séquences de réduction dans l'ARS représenté dans l'exemple 0.6 sont

$$\begin{aligned} \rho_2 \rho_3 \rho_4 &: o_1 \rightarrow o_4 \\ \rho_2 \rho_3 \rho_5 &: o_1 \rightarrow o_4 \end{aligned}$$

La séquence $\rho_1 \rho_3 \rho_4$ en revanche n'est pas une séquence de réduction à cause de la non correspondance entre $\text{tgt}(\rho_1) = o_2$ et $\text{src}(\rho_3) = o_3$.

Dans tout système abstrait de réécriture nous pouvons considérer une relation binaire sur les objets qui correspond à l'existence d'un pas de réduction entre deux objets.

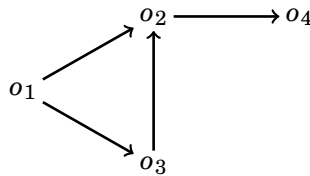
Définition 0.9 (Relation de réduction). Soit $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ un ARS. La **relation de réduction** de Σ , notée \rightarrow_Σ , est la plus petite relation binaire sur \mathcal{O} telle que pour tout pas de réduction $\rho : o \rightarrow o' \in \mathcal{R}$ nous avons $o \rightarrow_\Sigma o'$. La **clôture réflexive-transitive** de \rightarrow_Σ est notée $\twoheadrightarrow_\Sigma$.

Remarquons que la clôture réflexive-transitive $\twoheadrightarrow_\Sigma$ vérifie $o \twoheadrightarrow_\Sigma o'$ si et seulement s'il existe une séquence de réduction éventuellement vide $\rho : o \rightarrow o'$ dans Σ . Pour \rightarrow_Σ comme pour $\twoheadrightarrow_\Sigma$, l'indice pourra être omis en l'absence d'ambiguïté.

La relation de réduction d'un système abstrait de réécriture peut être vue comme un graphe dirigé simple.

Exemple 0.10.

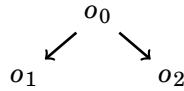
Le multigraphe de l'exemple 0.6 définit la relation de réduction dessinée ci-dessous. Dans ce graphe simple les noms des arcs n'existent plus, et en particulier les arcs parallèles ρ_4 et ρ_5 sont confondus.



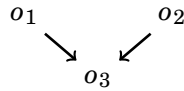
Les propriétés de *terminaison* et de *confluence* sont des critères généraux assurant l'existence et l'unicité du résultat d'un calcul [Ter03, Chap.1].

Définition 0.11 (Terminaison). Soit $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ un ARS. Une **forme normale** de Σ est un objet $o \in \mathcal{O}$ tel qu'il n'existe pas de réduction $\rho \in \mathcal{R}$ avec $\text{src}(\rho) = o$. Un objet $o \in \mathcal{O}$ est **faiblement normalisant**, ou **termine faiblement**, s'il existe une séquence de réduction de o à une forme normale. Un objet o est **fortement normalisant**, ou **termine**, s'il n'existe pas de séquence de réduction infinie depuis o . L'ARS Σ est **faiblement normalisant** (resp. **fortement normalisant**) si tous les objets de \mathcal{O} sont faiblement normalisants (resp. fortement normalisants).

Définition 0.12 (Propriété du diamant, confluence). Une relation binaire \rightarrow sur un ensemble \mathcal{O} a la **propriété du diamant** si pour tous $o_0, o_1, o_2 \in \mathcal{O}$ tels que



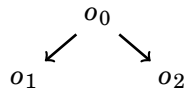
il existe $o_3 \in \mathcal{O}$ tel que



Une relation binaire \rightarrow sur un ensemble \mathcal{O} est **confluente** si sa clôture transitive a la propriété du diamant. Un ARS a la propriété du diamant (resp. est confluente) si sa relation de réduction a la propriété du diamant (resp. est confluente).

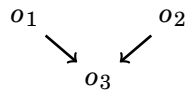
La propriété suivante de *confluence forte* correspond à une propriété du diamant dans laquelle l'un et l'autre des pas de réduction fermants peuvent être remplacés par une égalité. Cette caractérisation plus souple se révélera utile quand des pas de réduction pourront être *effacés*. Cette propriété est nommée *subcommutativity* dans Terese [Ter03, Chap. 1].

Définition 0.13 (Confluence forte). Une relation binaire \rightarrow sur un ensemble \mathcal{O} a la propriété de **confluence forte** si pour tous $o_0, o_1, o_2 \in \mathcal{O}$ tels que



l'une de ces trois propositions est vérifiée :

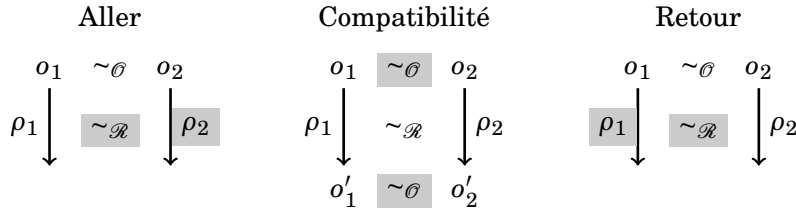
- $o_1 \longrightarrow o_2$
- $o_1 \longleftarrow o_2$
- il existe $o_3 \in \mathcal{O}$ tel que



Remarque 0.14. La confluence forte implique la confluence.

Nous pouvons exprimer que deux systèmes abstraits de réécriture sont équivalents à l'aide d'une notion de **bisimilarité** formalisant l'idée suivante : deux systèmes abstraits de réécriture sont bisimilaires si toute séquence de réduction de l'un quelconque de ces systèmes peut être mise en correspondance avec une séquence de réduction de l'autre, de manière à ce que chaque pas de réduction d'un côté corresponde à exactement un pas de réduction de l'autre côté.

Nous utiliserons la définition proposée par van Oostrom [Ter03, Chap. 8], qui se réduit à des relations sur les objets et sur les pas de réduction. Deux conditions (*Aller* et *Retour*) assurent que tout pas de réduction d'un côté a un pas de réduction correspondant de l'autre côté, et une troisième condition (*Compatibilité*) assure ensuite la bonne composition des séquences de pas de réduction correspondants. La situation est résumée par les dessins suivants, où sont coloriés les pas de réduction et relations imposés par ces conditions.



Définition 0.15 (Bisimulation). Soient deux ARS $\Sigma_1 = (\mathcal{O}_1, \mathcal{R}_1, \text{src}_1, \text{tgt}_1)$ et $\Sigma_2 = (\mathcal{O}_2, \mathcal{R}_2, \text{src}_2, \text{tgt}_2)$. Une **bisimulation** entre Σ_1 et Σ_2 est une paire $(\sim_{\mathcal{O}}, \sim_{\mathcal{R}})$ où $\sim_{\mathcal{O}}$ (resp. $\sim_{\mathcal{R}}$) est une relation entre \mathcal{O}_1 et \mathcal{O}_2 (resp. \mathcal{R}_1 et \mathcal{R}_2) vérifiant les propriétés suivantes.

Aller Soient deux objets $o_1 \in \mathcal{O}_1$ et $o_2 \in \mathcal{O}_2$ tels que $o_1 \sim_{\mathcal{O}} o_2$. Pour tout pas de réduction $\rho_1 \in \mathcal{R}_1$ tel que $\text{src}_1(\rho_1) = o_1$ il existe un pas de réduction $\rho_2 \in \mathcal{R}_2$ tel que $\text{src}_2(\rho_2) = o_2$ et $\rho_1 \sim_{\mathcal{R}} \rho_2$.

Retour Soient deux objets $o_1 \in \mathcal{O}_1$ et $o_2 \in \mathcal{O}_2$ tels que $o_1 \sim_{\mathcal{O}} o_2$. Pour tout pas de réduction $\rho_2 \in \mathcal{R}_2$ tel que $\text{src}_2(\rho_2) = o_2$ il existe un pas de réduction $\rho_1 \in \mathcal{R}_1$ tel que $\text{src}_1(\rho_1) = o_1$ et $\rho_1 \sim_{\mathcal{R}} \rho_2$.

Compatibilité Soient deux pas de réduction $\rho_1 \in \mathcal{R}_1$ et $\rho_2 \in \mathcal{R}_2$. Si $\rho_1 \sim_{\mathcal{R}} \rho_2$, alors $\text{src}_1(\rho_1) \sim_{\mathcal{O}} \text{src}_2(\rho_2)$ et $\text{tgt}_1(\rho_1) \sim_{\mathcal{O}} \text{tgt}_2(\rho_2)$.

S'il existe une bisimulation entre les ARS Σ_1 et Σ_2 , alors ces deux systèmes sont dits **bisimilaires**.

Dans les systèmes de réécriture que nous serons amenés à considérer, les différents pas de réduction ne seront pas tous indépendants les uns des autres. En particulier, après un pas de réduction ρ il restera souvent dans la cible $\text{tgt}(\rho)$ des survivances des autres pas de réduction qui étaient possibles depuis la source $\text{src}(\rho)$. Une relation de *résiduation* va permettre de suivre ces survivances, ou *résidus*, le long des séquences de réduction. Nous nous référerons aux définitions utilisées par Glauert et Khasidashvili [GK96].

Définition 0.16 (Résiduation, création). Soit $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ un ARS. Une opération de **résiduation** pour Σ est une fonction partielle $/$ de $\mathcal{R} \times \mathcal{R}$ dans les parties de \mathcal{R} telle que, pour tout pas de réduction $\rho_1 : o \rightarrow o' \in \mathcal{R}$:

- ρ_2/ρ_1 est défini si et seulement si $\text{src}(\rho_2) = \text{src}(\rho_1) = o$.
- Pour tout pas de réduction ρ_2 de source o , ρ_2/ρ_1 est un ensemble de pas de réduction de source $\text{tgt}(\rho_1) = o'$.
- Pour tous pas de réduction co-initiaux distincts ρ_2 et ρ_3 nous avons $(\rho_2/\rho_1) \cap (\rho_3/\rho_1) = \emptyset$.
- $\rho_1/\rho_1 = \emptyset$.

Soit un pas de réduction $\rho_1 : o \rightarrow o' \in \mathcal{R}$. Pour tous pas de réduction ρ_2 et ρ_3 , si $\rho_3 \in \rho_2/\rho_1$ alors ρ_3 est appelé un **résidu** de ρ_2 après ρ_1 . Pour tout pas de réduction ρ_3 , si ρ_3 n'est un résidu d'aucun pas de réduction ρ_2 de source t alors le pas de réduction ρ_3 est dit **créé** par ρ_1 .

La notion de résidus d'un pas de réduction ρ_2 après un pas de réduction ρ_1 s'étend naturellement aux résidus d'un ensemble de pas de réduction co-initiaux R_2 après une séquence de réduction ρ_1 :

- $R_2/\rho_1 = \bigcup_{\rho_2 \in R_2} \rho_2/\rho_1$
- $R_2/\rho_1\rho_1 = (R_2/\rho_1)/\rho_1$

Avec la définition précédente nous désignons comme résidus d'un pas de réduction un ensemble de pas de réduction co-initiaux, c'est-à-dire une multitude d'actions concurrentes dont il faut faire la synthèse pour vraiment comprendre ce qu'est devenu le pas de réduction d'origine. La notion de *développement* d'un ensemble R de pas de réduction co-initiaux permet de caractériser les séquences de réduction réalisant les pas de réduction de R .

Définition 0.17 (Développement). Soient $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ un ARS et $R \subseteq \mathcal{R}$ un ensemble de pas de réduction co-initiaux dans Σ dont nous notons o la source commune. Un **développement** de R est une séquence de réduction $\rho = \rho_1 \dots \rho_n$ de source o telle que pour tout $i \in \{1, \dots, n\}$ nous ayons $\rho_i \in R/\rho_1 \dots \rho_{i-1}$. Un **développement complet** de R est un développement ρ de R tel que $R/\rho = \emptyset$.

Remarquons que la séquence de réduction vide est l'unique développement de l'ensemble de réductions vide.

Un ensemble R de pas de réduction co-initiaux ayant de nombreux développements possibles, la synthèse de ces développements est en général équivoque. Les systèmes de résiduation déterministes (*Deterministic Residual Structures, DRS* [GK96]) décrivent un cadre dans lequel cette équivoque disparaît.

Définition 0.18 (Système de résiduation déterministe, DRS). Un **DRS** est une paire $(\Sigma, /)$ où $\Sigma = (\mathcal{O}, \mathcal{R}, \text{src}, \text{tgt})$ est un ARS et $/$ une opération de résiduation pour Σ vérifiant les axiomes suivants :

Développements finis Soit $R \subseteq \mathcal{R}$ un ensemble de pas de réduction co-initiaux de Σ , alors tous les développements de R terminent. De plus, tous les développements complets de R ont même cible et engendrent la même relation de résiduation.

Acyclicité Pour tous deux pas de réduction co-initiaux distincts $\rho, \rho' \in \mathcal{R}$, si $\rho/\rho' = \emptyset$ alors $\rho'/\rho \neq \emptyset$.

Dans ce cadre, pour tout ensemble $R \subseteq \mathcal{R}$ de pas de réduction co-initiaux nous notons $\text{dev}(R)$ un développement complet arbitraire de R .

0.3 Signatures, termes, contextes

Les systèmes de réécriture considérés dans cette thèse seront essentiellement des systèmes dont les objets sont des *termes*, c'est-à-dire des objets en forme d'arbres dont les nœuds sont décorés. Cette section rappelle les définitions générales et fixe les notations utilisées pour la manipulation des termes.

Définition 0.19 (Signature). Une **signature** \mathcal{S} est un ensemble au plus dénombrable muni d'une fonction d'**arité** associant un entier naturel à chaque élément de la signature.

Convention 0.20 (Signature). Une signature \mathcal{S} sera généralement décomposée sous la forme $\mathcal{S} = \bigcup_{n \in \mathbb{N}} \mathcal{S}_n$, où \mathcal{S}_n est l'ensemble des éléments de \mathcal{S} d'arité n .

Définition 0.21 (Termes syntaxiques). Soit \mathcal{S} une signature. Les **termes syntaxiques** sur \mathcal{S} sont les objets librement générés par la grammaire suivante :

$$t ::= a \mid f(t_1, \dots, t_n)$$

où $a \in \mathcal{S}_0$, $n \geq 1$ et $f \in \mathcal{S}_n$.

Les *contextes* représentent des termes partiels, dont les *trous* peuvent être remplis par d'autres contextes ou termes.

Définition 0.22 (Contextes). Un **contexte** sur une signature \mathcal{S} est un terme syntaxique sur la signature $\mathcal{S} \cup \{\square\}$ où \square est un symbole distingué figurant un **trou**. L'**arité** d'un contexte est le nombre d'occurrences du symbole distingué \square dans sa construction. Un contexte d'arité 0 est en particulier un terme syntaxique.

Définition 0.23 (Remplacement). Soit \mathcal{S} une signature, c un contexte d'arité n sur \mathcal{S} , et c_1, \dots, c_n n contextes d'arités arbitraires (qui peuvent donc en particulier être des termes syntaxiques) sur \mathcal{S} . Le **remplacement** $c[c_1, \dots, c_n]$ est un contexte dont l'arité est la somme des arités des c_i , qui est obtenu en remplaçant les n trous de c respectivement par les n contextes c_1, \dots, c_n . Le contexte c est appelé un **préfixe** du contexte $c[c_1, \dots, c_n]$.

Exemple 0.24.

$$\begin{aligned} @(\lambda x(\square), y)[@(x, y)] &= @(\lambda x(@(x, y)), y) \\ f(f(\square, b), \square)[a, b] &= f(f(a, b), b) \\ f(\square, \square)[a, f(\square, b)] &= f(a, f(\square, b)) \end{aligned}$$

La notion de position dans un terme permet d'identifier chaque nœud de l'arbre sous-jacent, via le chemin à suivre pour y accéder depuis la racine. Ceci donne un moyen non ambigu de désigner un symbole ou un sous-terme d'un terme donné, et permet également de spécifier des modifications d'un terme.

Définition 0.25 (Positions). Une **position** p est une séquence éventuellement vide d'entiers strictement positifs. Notons ϵ la position vide et $p \cdot q$ la **concaténation** de deux positions p et q . La concaténation s'étend aux ensembles de positions : $P \cdot Q = \{p \cdot q \mid p \in P, q \in Q\}$, et $p \cdot Q$ (resp. $P \cdot q$) abrègé $\{p\} \cdot Q$ (resp. $P \cdot \{q\}$).

Une position p est un **préfixe** d'une position q , noté $p < q$, s'il existe une position p' telle que $q = p \cdot p'$. Le préfixe est **strict** si de plus $p' \neq \epsilon$. Deux positions p et q sont **disjointes** quand aucune n'est un préfixe de l'autre.

Un **segment initial** est un ensemble de positions P tel que pour toutes positions $q < p$, si $p \in P$ alors $q \in P$.

L'ensemble des positions $\text{pos}(t)$ d'un terme syntaxique t est défini par :

$$\begin{aligned} \text{pos}(a) &= \{\epsilon\} \\ \text{pos}(f(t_1, \dots, t_n)) &= \{\epsilon\} \cup \bigcup_{1 \leq i \leq n} i \cdot \text{pos}(t_i) \end{aligned}$$

Pour tout terme t et toute position $p \in \text{pos}(t)$, notons $t(p)$ le **symbole** à la position p dans t , et $t|_p$ le **sous-terme** de t à la position p , tous deux définis comme suit :

$$\begin{array}{l|l} a(\epsilon) = a & a|_\epsilon = a \\ f(t_1, \dots, t_n)(\epsilon) = f & f(t_1, \dots, t_n)|_\epsilon = f(t_1, \dots, t_n) \\ f(t_1, \dots, t_n)(i \cdot p) = t_i(p) & f(t_1, \dots, t_n)|_{i \cdot p} = t_i|_p \end{array}$$

Exemple 0.26 (λ -termes). _____

Soit \mathcal{X} un ensemble dénombrable de variables. Considérons l'abstraction λx comme un symbole unaire et donnons au λ -calcul la signature $\mathcal{S}_\Lambda = \mathcal{X} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{\@\}$. Soit $t = @(\lambda x(@ (x, y)), y)$. Nous avons alors $\text{pos}(t) = \{\epsilon, 1, 11, 111, 112, 2\}$, $t(1) = \lambda x$, et $t|_{11} = @ (x, y)$.

Définition 0.27 (Remplacement parallèle). Pour tous termes t et u et tout ensemble de positions deux à deux disjointes $P \subseteq \text{pos}(t)$, notons $t[u]_P$ le **remplacement parallèle** défini par :

$$\begin{aligned} t[u]_\emptyset &= t \\ t[u]_{\{\epsilon\}} &= u \\ f(t_1, \dots, t_n)[u]_P &= f(t_1[u]_{P_1}, \dots, t_n[u]_{P_n}) \quad \text{où } P = \bigcup_i i \cdot P_i. \end{aligned}$$

Exemple 0.28 (Remplacement parallèle). _____

$$f(f(a, a), a)[b]_{\{12, 2\}} = f(f(a, b), b)$$

0.4 Réécriture du premier ordre

La réécriture du premier ordre offre des mécanismes de base pour la transformation de termes. Nous n'en verrons ici que quelques aspects basiques utiles à cette thèse. Pour le reste, Terese est une référence [Ter03].

Fixons un ensemble dénombrable \mathcal{Z}_0 d'éléments d'arité 0 appelés **méta-variables**. Nous utiliserons ces méta-variables pour construire les *méta-termes* qui permettent de définir les *règles de réécriture* qui déterminent les *pas de réduction* sur les termes « ordinaires ».

Définition 0.29 (TRS-termes). *Soit une signature \mathcal{S} . Les **termes** sur \mathcal{S} sont les termes syntaxiques sur la signature \mathcal{S} au sens de la définition 0.21. Les **méta-termes** sur \mathcal{S} sont les termes syntaxiques sur la signature $\mathcal{S} \sqcup \mathcal{Z}_0$. Un **motif** est un méta-terme qui n'est pas une méta-variable seule.*

Définition 0.30 (Valuation). *Une **valuation** σ associe des termes à des méta-variables. Notons t^σ le résultat obtenu en remplaçant chaque méta-variable de t par le terme indiqué par σ . Formellement :*

$$\begin{aligned} Z^\sigma &= \sigma(Z) & Z \in \mathcal{Z}_0 \\ a^\sigma &= a & a \in \mathcal{S} \\ f(t_1, \dots, t_n)^\sigma &= f(t_1^\sigma, \dots, t_n^\sigma) & f \in \mathcal{S} \end{aligned}$$

Définition 0.31 (Règles de réécriture). *Une **règle de réécriture** est une paire $\kappa : l \rightarrow r$ de méta-termes tels que l est un motif, et toutes les méta-variables de r apparaissent dans l .*

Définition 0.32 (TRS). *Un TRS est une paire $\Sigma = (\mathcal{S}, \mathcal{R})$ dans laquelle \mathcal{S} est une signature et \mathcal{R} est un ensemble de règles de réécriture.*

Définition 0.33 (Réduction). *Un **pas de réduction** ρ dans un TRS $\Sigma = (\mathcal{S}, \mathcal{R})$ est l'application d'une règle $\kappa : l \rightarrow r \in \mathcal{R}$ avec une valuation σ , dans un contexte c . Un tel pas de réduction est noté $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et nous appelons l^σ un **radical** et r^σ son **réduit**. La notation $\rho : t \xrightarrow{p} t'$ signifie que le radical se trouve à la position p dans t .*

La notion d'orthogonalité caractérise les TRS dont les règles ne peuvent pas interférer. Les TRS orthogonaux forment une sous-classe des TRS très étudiée, en particulier pour ses bonnes propriétés de confluence ou de permutation des pas de réduction.

Définition 0.34 (Orthogonalité). *Soit $\Sigma = (\mathcal{S}, \mathcal{R})$ un TRS. Le système Σ est **sans recouvrement** si les deux points suivants sont vérifiés.*

1. *Si un radical $l_1^{\sigma_1}$ pour une règle κ_1 contient un radical $l_2^{\sigma_2}$ pour une règle $\kappa_2 \neq \kappa_1$, alors il existe une méta-variable Z_1 dans le motif l_1 telle que le radical $l_2^{\sigma_2}$ est déjà contenu dans $\sigma_1(Z_1)$.*
2. *De même si un radical pour une règle κ_1 contient strictement un autre radical pour la même règle κ_1 .*

Le système Σ est **linéaire gauche** si aucun membre gauche de règle ne contient deux occurrences de la même méta-variable. Le système Σ est **orthogonal** s'il est à la fois linéaire gauche et sans recouvrement.

La notion suivante de descendance permet d'établir une relation entre les positions de la source et de la cible d'un pas de réduction. Dans les TRS orthogonaux elle permet aussi de définir un système de résiduation déterministe DRS.

Définition 0.35 (Descendance). *Considérons un pas de réduction $\rho : t = c[l^\sigma] \xrightarrow{q} c[r^\sigma] = t'$ et une position p de t . Les **descendants** de p après ρ forment l'ensemble p/ρ de positions de t' défini ainsi :*

- Si p et q sont disjointes alors $p/\rho = \{p\}$.
- Si p est un préfixe strict de q alors $p/\rho = \{p\}$.
- Si $p = q \cdot p_Z \cdot p_\sigma$ avec p_Z une position d'une méta-variable Z dans l et p_σ une position de $\sigma(Z)$, alors soit P_Z l'ensemble des positions des occurrences de Z dans r , $p/\rho = q \cdot P_Z \cdot p_\sigma$.
- Dans tous les autres cas $p/\rho = \emptyset$.

Si $p' \in p/\rho$ alors on appelle p un **ancêtre** de p' . La notion de descendant s'étend immédiatement des positions aux sous-termes.

0.5 Lambda-calcul

Le λ -calcul a été introduit il y a bien longtemps par Church [Chu32] et a connu un nouvel essor avec l'étude des langages de programmation. Cette section n'en rappelle que quelques éléments utiles au développement de cette thèse. Pour un exposé plus complet, se référer au livre de Barendregt [Bar84].

Définition 0.36 (λ -termes syntaxiques). *Soit \mathcal{X} un ensemble dénombrable dont les éléments seront appelés des **variables**. Les **λ -termes syntaxiques** sont donnés par la grammaire :*

$$t ::= x \mid \lambda x.t \mid tt \quad x \in \mathcal{X}$$

Convention 0.37. *Nous considérerons les λ -termes syntaxiques comme des termes syntaxiques sur la signature \mathcal{S} telle que :*

- $\mathcal{S}_0 = \mathcal{X}$
- $\mathcal{S}_1 = \{\lambda x \mid x \in \mathcal{X}\}$
- $\mathcal{S}_2 = \{@\}$

Les constructions $\lambda x.t$ et tt du λ -calcul seront considérées comme des notations respectivement pour $\lambda x(t)$ et $@(t,t)$. Nous héritons ainsi les notions de contextes, de remplacements, et de positions.

Dans les termes du λ -calcul, tout sous-terme de la forme $\lambda x.t$ établit un lien entre la λ -abstraction λx et certaines occurrences de la variable x dans t . Ce lien est régi par les notions suivantes de variables libres et de variables liées, qui sont exprimées ici dans le cadre général des contextes et non seulement des termes.

Définition 0.38 (Variables libres, variables liées). *Les ensembles $\text{fv}(c)$ et $\text{bv}(c)$ des **variables libres** (free variables) et **variables liées** (bound variables) d'un*

λ -contexte d'arité arbitraire c sont définis par les équations :

$$\begin{array}{ll} \text{fv}(x) = \{x\} & \text{bv}(x) = \emptyset \\ \text{fv}(\lambda x.c) = \text{fv}(c) \setminus \{x\} & \text{bv}(\lambda x.c) = \{x\} \cup \text{bv}(c) \\ \text{fv}(c_1 c_2) = \text{fv}(c_1) \cup \text{fv}(c_2) & \text{bv}(c_1 c_2) = \text{bv}(c_1) \cup \text{bv}(c_2) \end{array}$$

Un λ -contexte d'arité arbitraire c est **clos** quand $\text{fv}(c) = \emptyset$.

Étant donnée une variable libre d'un certain terme (ou contexte), nous pouvons extraire les positions des occurrences de cette variable qui sont libres.

Définition 0.39 (Occurrences libres). *L'ensemble $\text{fo}_x(c)$ des **occurrences libres** (free occurrences) d'une variable x dans un λ -contexte d'arité arbitraire c est le sous-ensemble des positions de c défini par les équations :*

$$\begin{array}{ll} \text{fo}_x(x) = \{\epsilon\} & \\ \text{fo}_x(y) = \emptyset & x \neq y \\ \text{fo}_x(\lambda x.c) = \emptyset & \\ \text{fo}_x(\lambda y.c) = 1 \cdot \text{fo}_x(c) & x \neq y \\ \text{fo}_x(c_1 c_2) = (1 \cdot \text{fo}_x(c_1)) \cup (2 \cdot \text{fo}_x(c_2)) & \end{array}$$

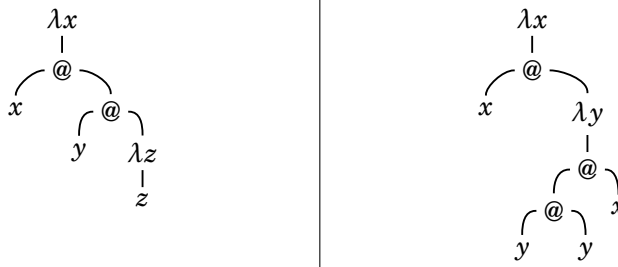
Remarque 0.40. *Nous avons $\text{fo}_x(c) \neq \emptyset$ si et seulement si $x \in \text{fv}(c)$.*

Définition 0.41 (Convention de Barendregt [Bar84]). *Un contexte d'arité arbitraire c respecte la convention de Barendregt, noté $\mathbb{B}(c)$, si tout sous-terme c' de c vérifie $\text{fv}(c') \cap \text{bv}(c') = \emptyset$.*

Définition 0.42 (Expressions libres [Wad71, PJ87]). *Une **expression libre** d'un λ -terme syntaxique t est un sous-terme strict s tel que toute occurrence de variable libre dans s est également libre dans t . Une expression libre de t est **maximale** si elle n'est pas un sous-terme d'une autre expression libre de t .*

Exemple 0.43.

Considérons le λ -terme $t = \lambda x.x(y\lambda z.z)$ (à gauche ci-dessous). Le sous-terme y est une expression libre de t , puisque cette variable n'est pas liée. Le sous-terme z n'est pas une expression libre de t , puisque cette variable est liée par la λ -abstraction λz . En revanche, le sous-terme z appartient au sous-terme $\lambda z.z$ qui est un sous-terme clos et donc une expression libre. Enfin, nous avons une expression libre maximale dans t , qui est $y\lambda z.z$.



Le λ -terme $u = \lambda x.x(\lambda y.yyx)$ (à droite ci-dessus) ne contient pas d'expression libre. En effet le sous-terme $\lambda y.yyx$ contient une occurrence libre de la variable x , et tout ses sous-termes contiennent au moins une occurrence libre de x ou de y .

Une opération de base du λ -calcul, définie au niveau méta, est la substitution d'un terme extérieur aux occurrences libres d'une certaine variable.

Définition 0.44 (Substitution). *La substitution $t^{x:=u}$ d'un terme u pour une variable x dans un terme t est définie par :*

$$\begin{aligned} x^{x:=u} &= u \\ y^{x:=u} &= y && \text{si } x \neq y \\ (\lambda y.t)^{x:=u} &= \lambda y.t^{x:=u} && \text{si } x \neq y \text{ et } y \notin \text{fv}(u) \\ (t_1 t_2)^{x:=u} &= t_1^{x:=u} t_2^{x:=u} \end{aligned}$$

Remarque 0.45. *Par la condition imposée dans le troisième cas, une substitution peut ne pas être définie. En revanche, si la substitution $t^{x:=u}$ est définie alors $t^{x:=u} = t[u]_{\text{fo}_x(t)}$.*

En λ -calcul les noms des variables liées n'ont pas de réelle signification ; ils ne servent qu'à définir les liens entre les λ -abstractions et les occurrences de variables. Par conséquent un nom peut être troqué pour un autre dans un λ -terme sans que cela n'affecte son comportement. La notion d' α -conversion capture ces modifications possibles.

Définition 0.46 (Alpha-conversion). *L' α -conversion est la congruence sur les λ -termes syntaxiques générée par l'équation :*

$$\lambda x.t =_\alpha \lambda y.t^{x:=y} \quad \text{si } y \notin \text{fv}(t)$$

Convention 0.47 (Lambda-termes). *Il est usuel d'appeler λ -termes les classes d'équivalence de $=_\alpha$. Cependant, nous utiliserons plus souvent dans cette thèse les λ -termes syntaxiques. Ainsi, dorénavant nous appellerons **λ -termes** les λ -termes syntaxiques et **λ -termes modulo α** les classes d'équivalence de $=_\alpha$.*

Les pas de réduction du λ -calcul sont définis au niveau méta par une unique règle, appelée β , selon laquelle à chaque fois qu'une λ -abstraction est appliquée à un deuxième λ -terme ce dernier peut être substitué dans le corps de ladite λ -abstraction. Ce mécanisme est réputé représenter l'application d'une fonction à un argument.

Définition 0.48 (Bêta-réduction). *La règle de β -réduction, aussi appelée règle β , est :*

$$(\lambda x.t)u \rightarrow_\beta t^{x:=u}$$

Tout terme de la forme $(\lambda x.t)u$ est appelé un **pré- β -radical**, et la λ -abstraction $\lambda x.t$ est appelée **λ -abstraction principale** du pré-radical. Si de plus la substitution $t^{x:=u}$ est définie alors le terme $(\lambda x.t)u$ est appelé un **radical**.

Le relation de β -réduction est la congruence sur les λ -termes engendrée par la règle β . Alternativement, la relation de β -réduction peut être définie par la condition :

$$c[(\lambda x.t)u] \rightarrow_{\beta} c[t^{x:=u}] \quad \text{pour tout contexte unaire } c$$

ou par les règles d'inférences :

$$\begin{array}{c} r = (\lambda x.t)u \xrightarrow{r}_{\beta} t^{x:=u} \quad (\beta) \\ \\ \frac{t \xrightarrow{r}_{\beta} t'}{t \rightarrow_{\beta} t'} \quad (w) \quad \frac{t_1 \xrightarrow{r}_{\beta} t'_1}{t_1 t_2 \xrightarrow{r}_{\beta} t'_1 t_2} \quad (\mu) \quad \frac{t_2 \xrightarrow{r}_{\beta} t'_2}{t_1 t_2 \xrightarrow{r}_{\beta} t_1 t'_2} \quad (\nu) \quad \frac{t \xrightarrow{r}_{\beta} t'}{\lambda x.t \xrightarrow{r}_{\beta} \lambda x.t'} \quad (\xi) \end{array}$$

Cette notion de réduction définit un *ARS*, que la notion de *descendance* suivante permet d'équiper d'un système de résiduation déterministe *DRS*.

Pour suivre les effets d'un pas de β -réduction $\rho : c[(\lambda x.t)u] \rightarrow c[t^{x:=u}]$ nous pouvons établir des correspondances entre les positions de la source $c[(\lambda x.t)u]$ et la cible $c[t^{x:=u}]$. C'est l'objet de la notion de *descendance*, qui à toute position de la source associe un ensemble de positions de la cible. Cette notion de descendance est à la base de la définition d'un système de résiduation pour le λ -calcul.

Définition 0.49 (Descendance). *Considérons un pas de réduction à la position q $\rho : c[(\lambda x.t)u] \xrightarrow{\rho} c[t^{x:=u}]$ et une position p de $c[(\lambda x.t)u]$. Les **descendants** de p après ρ forment l'ensemble p/ρ de positions de $c[t^{x:=u}]$ défini ainsi :*

- Si p et q sont des positions disjointes, alors $p/\rho = \{p\}$.
- Si p est un préfixe strict de q , alors $p/\rho = \{p\}$.
- Si $p = q \cdot 11 \cdot p_t$ avec $p_t \in \text{pos}(t) \setminus \text{fo}_x(t)$, alors $p/\rho = \{q \cdot p_t\}$.
- Si $p = q \cdot 2 \cdot p_u$ avec p_u une position de u , alors $p/\rho = q \cdot (\text{fo}_x(t)) \cdot p_u$.
- Dans tous les autres cas $p/\rho = \emptyset$. Ces cas sont $p = q$, $p = q \cdot 1$ et $p \in q \cdot 11 \cdot \text{fo}_x(t)$.

Si $p' \in p/\rho$ alors nous appelons p un **ancêtre** de p' . La notion de descendant s'étend immédiatement des positions aux symboles et aux sous-termes. Un descendant r' d'un β -radical r est appelé un **résidu** de r . Un β -radical r' dans un terme t' est dit **créé** par un pas de β -réduction $\rho : t \rightarrow t'$ s'il n'est pas un résidu d'un β -radical de t .

Remarquons que nous employons ici le terme *résidu* pour qualifier un β -radical alors que la notion abstraite vue à la section 0.2 concerne des pas de réduction. Ceci est légitime dans ce cas puisque dans le λ -calcul une occurrence d'un radical dans un λ -terme source suffit à définir un pas de réduction.

Remarque 0.50 (Existence de l'ancêtre). *Soit une réduction $t \rightarrow t'$. Toute position p' de t' admet un ancêtre dans t .*

L'évaluation des programmes fonctionnels est un jeu de passage d'arguments à des fonctions, qui ne se soucie généralement pas d'évaluer le corps d'une fonction avant que cette dernière soit instanciée. Ceci nous enjoint à étudier les *réductions faibles*, dans lesquelles la réduction *sous une abstraction* est restreinte sinon interdite.

Définition 0.51 (Réductions faibles). *La β -réduction faible naïve est la relation de réduction obtenue en supprimant la règle d'inférence (ξ) . La β -réduction faible combinatoire [cH98] est la relation de réduction obtenue en remplaçant la règle d'inférence (ξ) par la règle contrainte (ξ') :*

$$\frac{t \xrightarrow{\beta} t' \quad x \notin \text{fv}(r)}{\lambda x.t \xrightarrow{\beta} \lambda x.t'} (\xi')$$

Remarque 0.52. *Contrairement à la β -réduction faible naïve, la β -réduction faible combinatoire est confluente [cH98].*

Convention 0.53. *Dans cette thèse, sauf mention contraire par β -réduction faible nous désignerons la β -réduction faible combinatoire.*

La réduction faible (combinatoire) peut être définie alternativement par l'identification des positions « gelées » auxquelles ne doit pas se trouver un radical. Ces positions sont celles comprises entre la position d'une occurrence de variable liée et la position du lieu associé.

Définition 0.54 (Positions gelées). *Soit t un λ -terme et $p \in \text{pos}(t)$. La position p est **gelée** s'il existe deux positions $q_\lambda, q_x \in \text{pos}(t)$, une variable x et un λ -terme u tels que $q_\lambda < p < q_x$, $t|_{q_\lambda} = \lambda x.u$, et $t|_{q_x}$ est une occurrence de x libre dans u .*

La β -réduction faible combinatoire autorise la réduction exactement des radicaux dont la position racine n'est pas gelée.

Remarque 0.55. *Les réductions faibles, en tant que cas particulier de réductions closes [FMS05], ne nécessitent pas de renommage de variables. Ainsi, dans l'étude de la réduction faible il sera pertinent de travailler avec des termes syntaxiques plutôt que modulo α .*

0.6 Combinatory Reduction Systems

Les *Combinatory Reduction Systems* (CRS) sont un format de réécriture de termes d'ordre supérieur créé par Klop [Klo80, KvOvR93] dans lequel la réécriture du premier ordre et le λ -calcul se plongent naturellement, ainsi que les combinaisons des deux. Des exemples fameux sont des λ -calculs avec motifs algébriques [KvOdV08], des opérations primitives comme `fold` ou `map` sur des structures de données inductives [Ter03, Chap. 11] ou des combinateurs de points fixes [KvOvR93].

Nous en utilisons ici une présentation essentiellement classique [KvOvR93], mais avec quelques définitions supplémentaires concernant une classification des positions.

Pour tout $n \in \mathbb{N}$ soit \mathcal{Z}_n un ensemble dénombrable dont on appelle les éléments **méta-variables d'arité n** . On note $\mathcal{Z} = \bigcup_n \mathcal{Z}_n$ l'ensemble des **méta-variables**. Soit \mathcal{X} un ensemble dénombrable dont on appelle les éléments **variables**.

Définition 0.56 (CRS-termes). Soit une signature \mathcal{S} . Les **méta-termes syntaxiques** sur \mathcal{S} sont définis par la grammaire suivante :

$$t ::= x \mid [x]t \mid f(t_1, \dots, t_n) \mid Z(t_1, \dots, t_1)$$

où x est une variable, et f (resp. Z) est un symbole de \mathcal{S} (resp. une méta-variable) d'arité n . Un **terme** t est un méta-terme sans méta-variable.

Convention 0.57 (Abstractions). Fixons pour toute cette thèse un symbole distingué \diamond , qui sera un constructeur unaire présent dans toute signature de CRS. Dans tout méta-terme syntaxique ce constructeur apparaîtra en association avec chaque abstraction sous la forme $\diamond([x]t)$, et il n'aura pas d'autre occurrence. La notation $\langle x \rangle t$ abrégera $\diamond([x]t)$. Dans les exemples nous abrégerons également $\langle x \rangle \langle y \rangle t$ (c'est-à-dire $\diamond([x] \diamond([y]t))$) par $\langle x, y \rangle t$.

Convention 0.58. Nous considérerons les méta-termes syntaxiques sur une signature \mathcal{S} comme des termes syntaxiques (au sens de la définition 0.21) sur la signature \mathcal{S}' telle que :

- $\mathcal{S}'_0 = \mathcal{X} \cup \mathcal{S}_0 \cup \mathcal{Z}_0$
- $\mathcal{S}'_1 = \{\langle x \rangle \mid x \in \mathcal{X}\} \cup \mathcal{S}_1 \cup \mathcal{Z}_1$
- $\mathcal{S}'_n = \mathcal{S}_n \cup \mathcal{Z}_n$ pour tout $n \geq 2$

La construction $\langle x \rangle t$ sera considérée comme une notation pour $\langle x \rangle(t)$. Nous héritons ainsi les notions de **contextes**, de **remplacements**, et de **positions**. Nous appellerons de plus **position propre** d'un méta-terme syntaxique t est une position de t qui n'est pas la position d'une méta-variable.

Exemple 0.59.

Considérons une signature \mathcal{S} dans laquelle f est un symbole d'arité 2 et g un symbole d'arité 1. Soit le méta-terme

$$l = f(Z_1, \langle x, y \rangle Z_2(x, y))$$

L'ensemble des positions du méta-terme l est

$$\text{pos}(l) = \{\epsilon, 1, 2, 21, 211, 2111, 2112\}$$

et nous avons par exemple $l(1) = Z_1$ et $l|_{21} = \langle y \rangle Z(x, y)$.

Définition 0.60 (Méta-variables). Nous notons $\text{mv}(t)$ l'ensemble des méta-variables d'un méta-terme t .

$$\begin{aligned} \text{mv}(x) &= \{x\} \\ \text{mv}(\langle x \rangle t) &= \text{mv}(t) \\ \text{mv}(f(t_1, \dots, t_n)) &= \bigcup_i \text{mv}(t_i) \\ \text{mv}(Z(t_1, \dots, t_n)) &= \{Z\} \cup (\bigcup_i \text{mv}(t_i)) \end{aligned}$$

Définition 0.61 (Variables liées). *La construction syntaxique $\langle x \rangle t$ lie la variable x dans t . Ceci correspond à la définition suivante pour les **variables libres** $\text{fv}(t)$ (free variables) et **variables liées** $\text{bv}(t)$ (bound variables) d'un terme t .*

$$\begin{aligned} \text{fv}(x) &= \{x\} & \text{bv}(x) &= \emptyset \\ \text{fv}(\langle x \rangle t) &= \text{fv}(t) \setminus \{x\} & \text{bv}(\langle x \rangle t) &= \{x\} \cup \text{bv}(t) \\ \text{fv}(f(t_1, \dots, t_n)) &= \bigcup_i \text{fv}(t_i) & \text{bv}(f(t_1, \dots, t_n)) &= \bigcup_i \text{bv}(t_i) \\ \text{fv}(Z(t_1, \dots, t_n)) &= \bigcup_i \text{fv}(t_i) & \text{bv}(Z(t_1, \dots, t_n)) &= \bigcup_i \text{bv}(t_i) \end{aligned}$$

Un méta-terme syntaxique t est **clos** quand $\text{fv}(t) = \emptyset$. Un **motif** t est un méta-terme syntaxique clos dont le constructeur principal est \diamond ou un symbole de \mathcal{S} et dans lequel les méta-variables ont comme arguments des variables liées deux à deux différentes.

Définition 0.62 (Expressions libres). *Une **expression libre** d'un terme syntaxique t est un sous-terme s tel que toute occurrence de variable libre dans s est également libre dans t . Une expression libre de t est **maximale** si elle n'est pas un sous-terme d'une autre expression libre de t .*

Définition 0.63 (Substitution). *On note $t^{x:=u}$ le méta-terme syntaxique t dans lequel u se substitue aux occurrences libres de x . Formellement :*

$$\begin{aligned} x^{x:=u} &= u \\ y^{x:=u} &= y & x &\neq y \\ (\langle y \rangle t)^{x:=u} &= \langle y \rangle (t^{x:=u}) & x &\neq y \text{ et } y \notin \text{fv}(u) \\ (f(t_1, \dots, t_n))^{x:=u} &= f(t_1^{x:=u}, \dots, t_n^{x:=u}) \\ (Z(t_1, \dots, t_n))^{x:=u} &= Z(t_1^{x:=u}, \dots, t_n^{x:=u}) \end{aligned}$$

Cette opération s'étend en la substitution simultanée de plusieurs variables.

Remarquons que la condition de la troisième équation fait que la substitution peut ne pas être définie.

Définition 0.64 (α -conversion). *L' **α -conversion** est la congruence sur les termes syntaxiques générée par l'équation :*

$$\langle x \rangle t =_\alpha \langle y \rangle t^{x:=y} \quad \text{si } y \notin \text{fv}(t)$$

Convention 0.65 (Termes). *Il est usuel d'appeler termes les classes d'équivalence de $=_\alpha$. Cependant, nous utiliserons plus souvent dans cette thèse les termes syntaxiques. Ainsi, dorénavant nous appellerons **termes** les termes syntaxiques et **termes modulo α** les classes d'équivalence de $=_\alpha$.*

Définition 0.66 (Substitut). *Un **substitut** n -aire est une fonction notée $\underline{\lambda}x_1 \dots x_n. t$ avec t un terme, qui prend n termes en entrée et retourne s'il est défini le terme*

$$(\underline{\lambda}x_1 \dots x_n. t)(t_1, \dots, t_n) = t^{x_1:=t_1, \dots, x_n:=t_n}$$

Une **expression libre** dans un substitut $\underline{\lambda}x_1 \dots x_n. t$ est une expression libre s de t (qui peut être t lui-même) telle que de plus $\text{fv}(s) \cap \{x_1, \dots, x_n\} = \emptyset$.

Une **position** dans un substitut $\underline{\lambda}x_1\dots x_n.t$ est une position de t qui n'est pas la position d'une occurrence de l'un des x_i . Une position p d'un substitut $\underline{\lambda}x_1\dots x_n.t$ est dite **statique** si elle appartient à une expression libre de $\underline{\lambda}x_1\dots x_n.t$, et **dynamique** sinon.

Exemple 0.67.

Considérons un symbole f d'arité 2, un symbole a d'arité 0, et le substitut

$$\underline{\lambda}x_1x_2.t = \underline{\lambda}x_1x_2.f(f(x_1,a),x_1)$$

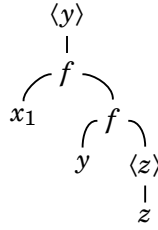
(où la duplication de x_1 et la non-occurrence de x_2 sont intentionnelles). L'ensemble des positions du substitut est

$$\text{pos}(\underline{\lambda}x_1x_2.t) = \{\epsilon, 1, 12\}$$

Dans cet ensemble les positions ϵ et 1 des deux occurrences du symbole f sont dynamiques tandis que la position 2 du symbole a est statique.

Considérons le substitut un peu plus compliqué

$$\underline{\lambda}x_1.u = \underline{\lambda}x_1.\langle y \rangle f(x_1, f(y, \langle z \rangle z))$$



L'ensemble de ses positions est

$$\underline{\lambda}x_1.u = \{\epsilon, 1, 12, 121, 122, 1221\}$$

Dans cet ensemble les positions $\epsilon, 1, 12, 121$ de l'abstraction $\langle y \rangle$, des deux occurrences du symbole f et de la variable y sont des positions dynamiques. En effet nous pouvons remarquer par exemple que le sous-terme $f(y, \langle z \rangle z)$ n'est pas une expression libre de u , ou que le sous-terme $u = \langle y \rangle f(x_1, f(y, \langle z \rangle z))$, qui est une expression libre de u , n'est pas une expression libre de $\underline{\lambda}x_1.u$ car il contient une occurrence de la variable x_1 . Les positions 122 et 1221 du sous-terme $\langle z \rangle z$ en revanche sont des positions statiques, car $\langle z \rangle z$ est une expression libre de $\underline{\lambda}x_1.u$.

Définition 0.68 (Valuation). Une **valuation** σ associe des substituts n -aires à des méta-variables n -aires. Notons t^σ le résultat obtenu en remplaçant chaque méta-variable de t par le substitut indiqué par σ . Formellement :

$$\begin{aligned} x^\sigma &= x \\ (\langle x \rangle t)^\sigma &= \langle x \rangle (t^\sigma) \\ (f(t_1, \dots, t_n))^\sigma &= f(t_1^\sigma, \dots, t_n^\sigma) \\ (Z(t_1, \dots, t_n))^\sigma &= \sigma(Z)(t_1^\sigma, \dots, t_n^\sigma) \end{aligned}$$

Une valuation σ est dite **sûre** s'il n'existe pas deux substituts $\sigma(Z)$ et $\sigma(Z')$ dans son codomaine et une variable x apparaissant libre dans un substitut et liée dans l'autre. Une valuation σ est dite **sûre** pour un méta-terme t si elle est sûre et s'il n'existe aucune méta-variable Z telle que $\sigma(Z)$ contient une occurrence libre d'une variable x qui est liée dans t .

Remarquons que l'application d'une valuation implique potentiellement des substitutions et peut dans certains cas ne pas être définie à cause de cela, indépendamment de sa sûreté. La sûreté ne fait en effet que prévenir les captures de variables qui ne sont pas déjà éliminées par les règles de substitution.

Exemple 0.69.

Considérons un symbole f d'arité 2, la valuation

$$\sigma = \{ Z \mapsto \underline{\lambda}x_1.\langle y \rangle f(x_1, f(y, z)) \}$$

et les trois méta-termes

$$\begin{aligned} t_1 &= \langle x \rangle Z(x) \\ t_2 &= \langle y \rangle Z(y) \\ t_3 &= \langle z \rangle Z(z) \end{aligned}$$

Le codomaine de la valuation σ ne contient qu'un substitut, elle est donc sûre. De plus, le codomaine de la valuation σ contient une unique variable libre z . Ainsi la valuation σ est sûre pour les termes t_1 et t_2 mais n'est pas sûre pour le terme t_3 . Remarquons toutefois que si la valuation σ est sûre pour le méta-terme t_2 , le méta-terme t_2^σ n'est pas défini, car la substitution $(\langle y \rangle f(x_1, f(y, z)))^{[x_1 := y]}$ n'est pas définie. Nous avons bien en revanche $t_1^\sigma = \langle x, y \rangle f(x, f(y, z))$.

Définition 0.70 (Positions résultantes). Soit t un méta-terme, σ une valuation pour les méta-variables de t , et p une position de t . Les **positions résultantes** de p dans t^σ forment un sous-ensemble $p^{t, \sigma}$ des positions de t^σ défini par :

$$\begin{aligned} e^{t, \sigma} &= \{e\} \\ (1 \cdot p)^{\langle x \rangle t, \sigma} &= 1 \cdot (p^{t, \sigma}) \\ (i \cdot p)^{f(t_1, \dots, t_n), \sigma} &= i \cdot (p^{t_i, \sigma}) \\ (i \cdot p)^{Z(t_1, \dots, t_n), \sigma} &= \{q \mid t|_q = x_i \text{ où } \sigma(Z) = \underline{\lambda}x_1 \dots x_n.t\} \cdot (p^{t_i, \sigma}) \end{aligned}$$

Une position résultante d'une position propre de t est appelée une **position propre** de t^σ . Toute position p de t^σ qui n'est pas une position propre de t^σ peut être décomposée en $p = p_Z \cdot p_\sigma$ avec p_Z une position résultante d'une position d'une méta-variable Z dans t et p_σ une position de $\sigma(Z)$. Dans ce cas, p est appelée une **position statique** (resp. **dynamique**) de t^σ si p_σ est une position statique (resp. dynamique) de $\sigma(Z)$. On note $\mathcal{P}_p(t, \sigma)$, $\mathcal{P}_s(t, \sigma)$ et $\mathcal{P}_d(t, \sigma)$ les ensembles des positions propres, statiques, et dynamiques de t^σ .

Exemple 0.71.

Considérons les symboles f , g , h , a , b et c d'arités respectives 2, 1, 2, 0, 0, et 0, et le méta-terme

$$r = h(Z_2(g(Z_1), b), Z_1)$$

Soit une valuation sûre pour r

$$\sigma = \{ Z_1 \mapsto \underline{\lambda}\phi.a, Z_2 \mapsto \underline{\lambda}x_1x_2.f(x_1, f(c, x_1)) \}$$

Nous avons

$$r^\sigma = h(f(g(a), f(c, g(a))), a)$$

La position ϵ du symbole h dans r a une position résultante après σ , qui est ϵ . La position 12 du symbole b dans r n'a pas de position résultante après σ . La position 11 du symbole g dans r a deux positions résultantes après σ , qui sont 11 et 122.

Dans le terme r^σ les positions ϵ , 11, et 122 des symboles g et h sont des positions propres, les positions 1 et 12 du symboles f sont des positions dynamiques, et les positions 111, 121, 1221, et 2 des symboles a et c sont des positions statiques.

Définition 0.72 (Règle de réécriture). Une **règle de réécriture** est une paire $\kappa : l \rightarrow r$ de méta-termes clos tels que l est un motif, et toutes les méta-variables de r apparaissent dans l . Une **variante** d'une règle de réécriture $\kappa_0 : l_0 \rightarrow r_0$ est une règle de réécriture $\kappa : l \rightarrow r$ telle que $l =_{\alpha} l_0$ et $r =_{\alpha} r_0$.

Définition 0.73 (Valuation sûre pour une règle). Une valuation σ pour les méta-variables $\text{mv}(l)$ du membre gauche l d'une règle $\kappa : l \rightarrow r$ est **sûre** pour la règle κ si elle est sûre pour ses membres gauche et droit l et r (ce qui requiert également que la valuation σ soit elle-même sûre).

Définition 0.74 (CRS). Un CRS est une paire $\Sigma = (\mathcal{S}, \mathcal{R})$ dans laquelle \mathcal{S} est une signature et \mathcal{R} est un ensemble de règles de réécriture.

Définition 0.75 (Réduction). Un **pas de réduction** ρ dans un CRS $\Sigma = (\mathcal{S}, \mathcal{R})$ est l'application d'une variante $\kappa : l \rightarrow r$ règle $\kappa_0 \in \mathcal{R}$ avec une valuation σ sûre pour κ , dans un contexte c . Un tel pas de réduction est noté $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et nous appelons l^σ un **radical** et r^σ son **réduit**. La notation $\rho : t \xrightarrow{p} t'$ signifie que le radical se trouve à la position p dans t .

Exemple 0.76.

Considérons les symboles f, g, h, a, b , et c d'arités respectives 2, 1, 2, 0, 0, et 0.

- Le méta-terme $l = f(Z_1, \langle x, y \rangle Z_2(x, y))$ est un motif clos.
- Le méta-terme $r = h(Z_2(g(Z_1), b), Z_1)$ est clos.
- La paire $l \rightarrow r$ est une règle de réécriture.
- La valuation $\{ Z_1 \mapsto \underline{\lambda}\phi.a, Z_2 \mapsto \underline{\lambda}x_1x_2.f(x_1, f(c, x_1)) \}$ est sûre pour $l \rightarrow r$.
- Avec la règle $l \rightarrow r$, la valuation σ et le contexte vide nous avons le pas de réduction

$$\rho : f(a, \langle x, y \rangle f(x, f(c, x))) \rightarrow h(f(g(a), f(c, g(a))), a)$$

La notion d'orthogonalité caractérise les CRS dont les règles ne peuvent pas interférer. Les CRS orthogonaux ont notamment de bonnes propriétés vis-à-vis des permutations de pas de réduction, qui nous seront utiles pour parler de causalité et d'optimalité. Le λ -calcul est un système orthogonal.

Définition 0.77 (Orthogonalité). Soit $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS. Le système Σ est **sans recouvrement** si les deux points suivants sont vérifiés.

1. Si un radical $l_1^{\sigma_1}$ pour une règle κ_1 contient un radical $l_2^{\sigma_2}$ pour une règle $\kappa_2 \neq \kappa_1$, alors il existe une méta-variable $Z_1 \in \text{mv}(l_1)$ telle que le radical $l_2^{\sigma_2}$ est déjà contenu dans $\sigma_1(Z_1)$.
2. De même si un radical pour une règle κ_1 contient strictement un autre radical pour la même règle κ_1 .

Le système Σ est **linéaire gauche** si aucun membre gauche de règle ne contient deux occurrences de la même méta-variable. Le système Σ est **orthogonal** s'il est à la fois linéaire gauche et sans recouvrement.

La notion suivante de descendance permet d'établir une relation entre les positions de la source et de la cible d'un pas de réduction. Dans les CRS orthogonaux elle permet aussi comme pour le λ -calcul de définir un système de résiduation déterministe.

Définition 0.78 (Descendance). Considérons un pas de réduction $\rho : t = c[l^\sigma] \xrightarrow{q} c[r^\sigma] = t'$ et une position p de t . Les **descendants** de p après ρ forment l'ensemble p/ρ de positions de t' défini ainsi :

- Si p et q sont disjointes alors $p/\rho = \{p\}$.
- Si p est un préfixe strict de q alors $p/\rho = \{p\}$.
- Si $p = q \cdot p_Z \cdot p_\sigma$ avec p_Z une position résultante d'une position d'une méta-variable Z dans l et p_σ une position de $\sigma(Z)$, alors soit P_Z l'ensemble des positions des occurrences de Z dans r , $p/\rho = q \cdot (\bigcup_{p \in P_Z} p^{r,\sigma}) \cdot p_\sigma$.
- Dans tous les autres cas $p/\rho = \emptyset$.

Si $p' \in p/\rho$ alors on appelle p un **ancêtre** de p' . La notion de descendant s'étend immédiatement des positions aux sous-termes.

Exemple 0.79.

Considérons la règle $l \rightarrow r$, la valuation σ , et le pas de réduction $\rho : l^\sigma \rightarrow r^\sigma$ vus à l'exemple 0.76. Les positions ϵ , 2, et 21 de l^σ n'ont pas de descendant après ρ . La position 1 de l^σ a pour descendants 111, 1221, et 2.

PREMIÈRE PARTIE

PARTAGE

Introduction

Cette première partie, constituée du seul chapitre 1, introduit le formalisme qui sera utilisé dans tous les chapitres suivants pour la description de systèmes de réécriture avec partage.

Ce cadre des *systèmes de partage étiquetés* (*Sharing-via-Labeling Systems, SLS*) décrit la réécriture de graphes d'une manière qui est à la fois :

- indirecte, car le formalisme n'utilise que des termes et non explicitement des graphes, et
- transparente, car le passage d'un terme d'un système étiqueté au graphe qu'il représente est immédiat.

Les techniques utilisées ont pour origine directe les travaux de Maranget sur les réductions optimales dans les systèmes de réécriture du premier ordre orthogonaux [Mar91, Mar92]. Ces travaux sont généralisés et portés dans un cadre axiomatique abstrait qui permettra d'atteindre de nombreux systèmes de réécriture d'ordre supérieur, fussent-ils non-orthogonaux. Ce chapitre 1 a fait l'objet d'une publication en 2012 [Bal12a].

CHAPITRE 1

Un cadre axiomatique pour le partage

Sommaire

1.1	Systèmes abstraits de réécriture de termes	59
1.2	Des étiquettes pour un exemple informel	62
1.3	Le cadre axiomatique	65
1.4	Exemple formel	69
1.5	Comparaison d'étiquetages	71
1.6	Discussion	73

La réécriture de graphes [BvEG⁺87] présente une amélioration par rapport à la réécriture de termes traditionnelle en ce qu'elle permet de partager des sous-termes, et ainsi d'éviter la duplication de certaines étapes de calcul. Cependant, les formalismes de graphes sont généralement bien plus complexes que les formalismes de termes correspondants. Les graphes d'ordre supérieur en particulier introduisent des procédures de décodage et des critères de correction non-triviaux [Wad71, Kah98].

Ce chapitre présente un cadre général pour le partage de sous-termes, dans lequel le formalisme est aussi simple que la réécriture de termes. Ce travail est une généralisation du travail de Maranget [Mar91, Mar92], dont voici deux éléments clés :

- Les graphes sont représentés par des termes étiquetés.
- Les réductions de graphes sont simulées par des séquences de réductions de termes.

Ainsi, nous formalisons la réécriture de graphes en utilisant uniquement la réécriture de termes bien connue.

Les travaux dans cette direction ont commencé il y a 20 ans dans le cadre des systèmes de réécriture du premier ordre orthogonaux [Mar91, Mar92], et ont ensuite été étendus à divers λ -calculs faibles [BLM07, Bal12b] ou extensions orthogonales de ces calculs faibles [Bal10a]. Sa simplicité technologique a rendu cette approche utile en tant qu'outil d'analyse, en particulier en permettant la

description de stratégies de réductions optimales [Mar91, Bal10a] et en permettant des comparaisons simples entre plusieurs mises en œuvre partagées du λ -calcul [Bal12b].

Dans cette approche, chaque symbole d'un terme se voit attribuer une étiquette, qui figure un emplacement. Deux sous-termes avec la même étiquette sont ainsi censés être physiquement égaux, occupant le même espace en mémoire ou dessinés aux mêmes coordonnées dans un schéma (voir exemple 1.1). La réécriture dans un graphe est alors simulée par la réduction simultanée de tous les radicaux du terme étiqueté correspondant à un même radical du graphe, c'est-à-dire tous les radicaux du terme étiqueté portant une étiquette donnée.

Exemple 1.1.

Les deux occurrences de a^δ (soulignées) représentent le même nœud de graphe (celui marqué d'une étoile) et sont réduites simultanément. Nous utilisons une flèche simple \rightarrow pour désigner la réduction d'un unique radical, et une flèche double \Rightarrow pour la réduction simultanée de plusieurs radicaux.

$$\Rightarrow \begin{array}{l} f^\alpha(f^\beta(a^\gamma, \underline{a^\delta}), \underline{a^\delta}) \\ f^\alpha(f^\beta(a^\gamma, b^\zeta), b^\zeta) \end{array} \quad \left| \quad \begin{array}{c} \begin{array}{c} \curvearrowright f \\ a \quad \quad a^* \end{array} \\ \Rightarrow \\ \begin{array}{c} \curvearrowright f \\ a \quad \quad b \end{array} \end{array}$$

Cette correspondance a du sens tant que les systèmes préservent une *propriété de partage* sur leurs termes étiquetés : si deux sous-termes cohabitent avec une même étiquette, alors ils sont syntaxiquement égaux. Remarquons que dans ce cadre, un terme étiqueté t vérifiant la propriété de partage est lui-même un témoin de la correction du graphe correspondant \mathcal{G}_t , et que de plus \mathcal{G}_t est un *graphe acyclique*.

La réduction simultanée n'est pas en général une extension bénigne à la théorie de la réécriture usuelle, mais le cas particulier considéré ici peut facilement être défini dans le cadre habituel. En effet, nous considérons uniquement la réduction simultanée de sous-termes *disjoints*, ce que nous marquons par l'appellation *réduction parallèle*. Et cette réduction peut être simulée par la réduction séquentielle des différents radicaux concernés dans n'importe quel ordre.

Exemple 1.2.

La réduction parallèle de l'exemple 1.1 peut être implémentée au choix par l'une ou l'autre de ces séquences de deux pas de réduction, dans lesquelles les radicaux contractés sont soulignés :

$$\begin{array}{l} f^\alpha(f^\beta(a^\gamma, \underline{a^\delta}), a^\delta) \rightarrow f^\alpha(f^\beta(a^\gamma, b^\zeta), \underline{a^\delta}) \rightarrow f^\alpha(f^\beta(a^\gamma, b^\zeta), b^\zeta) \\ f^\alpha(f^\beta(a^\gamma, a^\delta), \underline{a^\delta}) \rightarrow f^\alpha(f^\beta(a^\gamma, \underline{a^\delta}), b^\zeta) \rightarrow f^\alpha(f^\beta(a^\gamma, b^\zeta), b^\zeta) \end{array}$$

Ce premier chapitre introduit le cadre axiomatique pour la représentation du partage par des termes étiquetés que nous utiliserons tout au long de cette thèse. Ce formalisme se veut à la fois :

Simple : il exprime la réécriture partagée sans requérir plus de technologie formelle que la réécriture de termes.

Expressif : il permet la description d'une variété de systèmes de graphes pour tout genre de système de réécriture de termes.

Utilisable : il repose sur des axiomes peu nombreux et simples à utiliser en s'aidant des exemples de plus en plus riches qui émaillent cette thèse.

La section 1.1 introduit un cadre abstrait pour la réécriture de termes. En section 1.2 nous construirons de toutes pièces un étiquetage pour un système exemple qui mettra en lumière les principes du cadre axiomatique défini formellement à la section suivante. Cette section 1.3 suivante fournit également la preuve du théorème principal du chapitre (théorème 1.19 de partage) qui assure que tout système de termes étiquetés vérifiant les axiomes décrit bien un système de réécriture de graphes. La section 1.4 traite formellement l'exemple de la section 1.2. La section 1.5 donne des outils pour la comparaison de plusieurs étiquetages d'un même système et la section 1.6 conclut par une discussion sur l'expressivité et l'utilisabilité du cadre axiomatique et sa comparaison avec d'autres techniques.

1.1 Systèmes abstraits de réécriture de termes

Les *systèmes abstraits de réécriture* (*Abstract Rewriting Systems, ARS*, [GK96, Mel96]) donnent une description la plus générale possible de la réécriture. Un système \mathcal{y} est représenté par seulement quatre éléments : un ensemble \mathcal{O} d'objets, un ensemble \mathcal{R} de pas de réduction entre ces objets, et deux fonctions $\text{src} : \mathcal{R} \rightarrow \mathcal{O}$ et $\text{tgt} : \mathcal{R} \rightarrow \mathcal{O}$ associant un objet source et un objet cible à chaque pas de réduction. Les *ARS* peuvent exprimer la réécriture de n'importe quel type d'objets au moyen de n'importe quel type de règles de réduction et de conditions d'application de ces règles.

Cette section introduit les *systèmes abstraits de réécriture de termes* (*Abstract Term Rewriting Systems, ATRS*, définition 1.3) comme une spécialisation des *ARS* au cadre de la réécriture de termes. Dans un *ATRS*, l'ensemble d'objets \mathcal{O} est un ensemble de termes, et tout pas de réduction ρ est censé ne modifier qu'un sous-terme de la source (voir figure 1.1). La position de ce sous-terme modifié par ρ est appelée racine de ρ et notée $\text{root}(\rho)$.

De manière équivalente à cette présentation proche des *ARS*, tout pas de réduction ρ peut être caractérisé par trois autres composants faisant plutôt référence aux cadres concrets de réécriture de termes (voir figure 1.2) : le radical $\text{dex}(\rho)$ (*redex*) est le sous-terme à modifier de la source, le réduit $\text{duct}(\rho)$ (*reduct*) est le sous-terme modifié de la cible, et le contexte $\text{ctx}(\rho)$ est le reste, c'est-à-dire une partie non modifiée commune à la source et à la cible.

À cette structure nous ajoutons des axiomes assurant la bonne forme des pas de réduction. Un premier axiome *Source & Cible* impose aux source $\text{src}(\rho)$ et cible $\text{tgt}(\rho)$ d'un pas de réduction ρ d'être dans l'ensemble de termes \mathcal{O} , que l'on peut interpréter comme un ensemble de termes *bien formés*. Un axiome *Résidus*

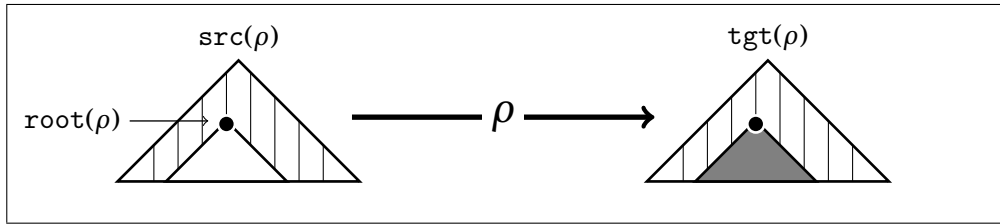
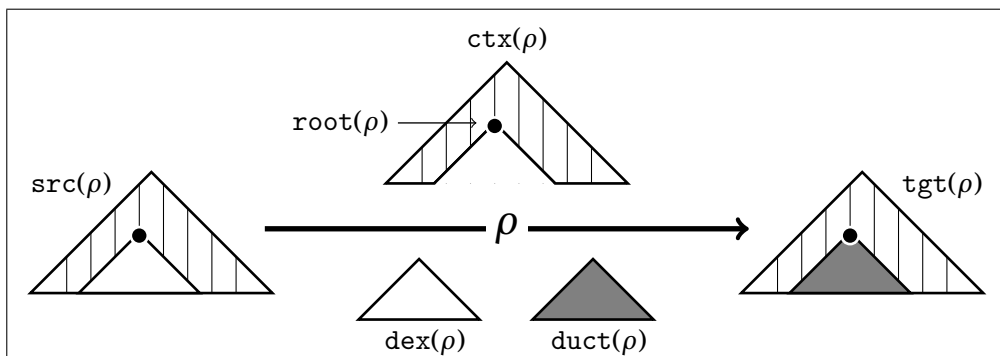
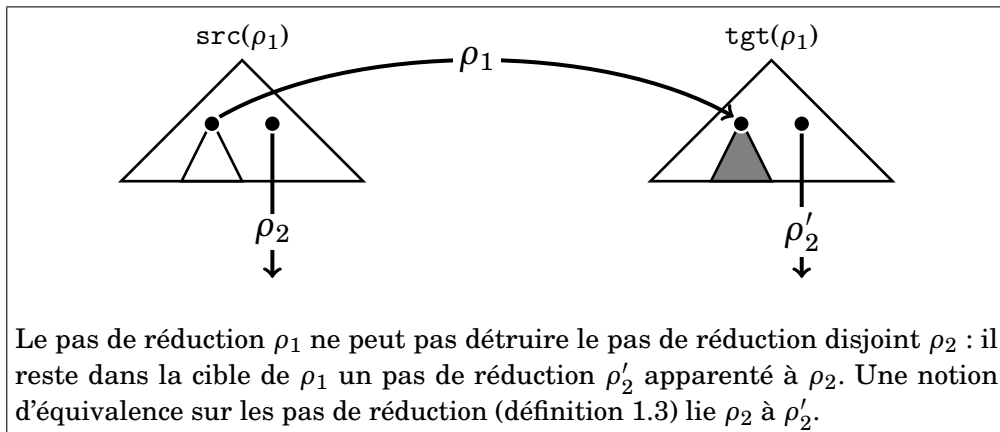


FIGURE 1.1: Réécriture abstraite de termes, présentation *ARS* : source, cible, et racine.



Remarquons que chacun des deux triplets $(src, tgt, root)$ et $(dex, duct, ctx)$ est suffisant pour déduire l'autre. Par exemple, décomposer la source au niveau de la racine permet de séparer le contexte du radical. Inversement, recoller le contexte et le réduit d'un pas de réduction en donne la cible.

FIGURE 1.2: Réécriture abstraite de termes, présentation *TRS* : radical, réduit, et contexte.



Le pas de réduction ρ_1 ne peut pas détruire le pas de réduction disjoint ρ_2 : il reste dans la cible de ρ_1 un pas de réduction ρ'_2 apparenté à ρ_2 . Une notion d'équivalence sur les pas de réduction (définition 1.3) lie ρ_2 à ρ'_2 .

FIGURE 1.3: Réductions à des positions disjointes

donne une contrepartie dynamique à l'observation qu'un pas de réduction n'a d'effet syntaxique que sur une partie délimitée du terme auquel il s'applique. Ce deuxième axiome interdit à des réductions affectant des sous-termes disjoints d'une même source d'interférer l'un avec l'autre (voir figure 1.3). Il est formalisé à l'aide d'une notion d'équivalence de pas de réduction caractérisant les pas de réduction qui ne diffèrent que par leur contexte. L'axiome *Résidus* assure en particulier que la réduction parallèle peut être simulée par des séquences de pas de réduction (lemme 1.8).

Définition 1.3 (ATRS). Soit $\Sigma = (\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx})$ un hexuplet tel que :

- \mathcal{S} est une signature.
- \mathcal{T} est un ensemble de termes sur \mathcal{S} .
- \mathcal{R} est un ensemble au plus dénombrable dont les éléments sont appelés **pas de réduction**.

et pour tout $\rho \in \mathcal{R}$:

- $\text{dex}(\rho)$ est un terme sur \mathcal{S} appelé **radical** de ρ .
- $\text{duct}(\rho)$ est un terme sur \mathcal{S} appelé **réduit** de ρ .
- $\text{ctx}(\rho)$ est un contexte unaire sur \mathcal{S} appelé **contexte** de ρ .

Pour tout pas de réduction $\rho \in \mathcal{R}$, la **source** (resp. la **cible**) de ρ est le terme $\text{src}(\rho) = \text{ctx}(\rho)[\text{dex}(\rho)]$ (resp. $\text{tgt}(\rho) = \text{ctx}(\rho)[\text{duct}(\rho)]$) et la **racine** de ρ est la position $\text{root}(\rho)$ de l'unique trou de $\text{ctx}(\rho)$.

Deux pas de réduction $\rho_1, \rho_2 \in \mathcal{R}$ sont dits **équivalents** si $\text{dex}(\rho_1) = \text{dex}(\rho_2)$ et $\text{duct}(\rho_1) = \text{duct}(\rho_2)$.

Pour tout terme $t \in \mathcal{T}$ et toute position $p \in \text{pos}(t)$, l'ensemble des pas de réduction $\rho \in \mathcal{R}$ tels que $\text{src}(\rho) = t$ et $\text{root}(\rho) = p$ est noté $\mathcal{R}(t, p)$.

Σ est un **ATRS** si les deux axiomes suivant sont vérifiés :

Source & Cible Pour tout pas de réduction $\rho \in \mathcal{R}$, $\text{src}(\rho) \in \mathcal{T}$ et $\text{tgt}(\rho) \in \mathcal{T}$.

Résidus Soient un pas de réduction $\rho_1 \in \mathcal{R}$ et une position $p \in \text{pos}(\text{src}(\rho_1))$ tels que les positions $\text{root}(\rho_1)$ et p sont disjointes. Alors pour tout pas de réduction $\rho_2 \in \mathcal{R}(\text{src}(\rho_1), p)$ il existe un pas de réduction $\rho_2' \in \mathcal{R}(\text{tgt}(\rho_1), p)$ équivalent à ρ_2 .

Définition 1.4 (ARS sous-jacent). Soit $\Sigma = (\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx})$ un ATRS. Lui sont associées comme ci-dessus des fonctions src et tgt de \mathcal{R} dans les termes sur \mathcal{S} . Ainsi le quadruplet $(\mathcal{T}, \mathcal{R}, \text{src}, \text{tgt})$ est un ARS que nous appellerons **ARS sous-jacent** de Σ . Cet ARS sous-jacent définit la **relation de réduction** de Σ .

Remarque 1.5. Remarquons que la définition 1.4 est cohérente car par axiome Source & Cible les images de src et tgt sont dans \mathcal{T} . En revanche le radical $\text{dex}(\rho)$ et le réduit $\text{duct}(\rho)$ d'un pas de réduction ρ dans un ATRS ne sont pas nécessairement dans l'ensemble \mathcal{T} des termes bien formés du système. Par axiome Source & Cible ils sont en revanche des sous-termes de termes de \mathcal{T} .

Dans tout ATRS est définie une notion de réduction parallèle combinant plusieurs pas de réduction disjoints en une étape.

Définition 1.6 (Réduction parallèle). Soient t un terme, et $\{\rho_1, \dots, \rho_n\}$ un ensemble de pas de réduction de source t dont les racines sont deux à deux disjointes. Soit

$$t' = t[\text{duct}(\rho_1)]_{\text{root}(\rho_1)} \dots [\text{duct}(\rho_n)]_{\text{root}(\rho_n)}$$

le terme obtenu à partir de t en remplaçant chaque radical par son réduit. Nous appelons **étape de réduction parallèle** l'action simultanée des pas de réduction ρ_1, \dots, ρ_n , notée $\bar{\rho} : t \Rightarrow t'$.

Exemple 1.7.

Soient ρ_1 et ρ_2 deux pas de réduction tels que :

$$\begin{array}{l|l} \text{dex}(\rho_1) = a & \text{src}(\rho_2) = f(f(a, a), a) \\ \text{duct}(\rho_1) = b & \text{tgt}(\rho_2) = f(f(a, a), b) \\ \text{ctx}(\rho_1) = f(f(a, _), a) & \text{root}(\rho_2) = 2 \end{array}$$

Alors $f(f(a, a), a) \Rightarrow f(f(a, b), b)$ par réduction parallèle de $\{\rho_1, \rho_2\}$.

Lemme 1.8 (Simulation de la réduction parallèle). Dans tout ATRS, si $t \Rightarrow t'$ alors $t \twoheadrightarrow t'$.

Démonstration. Par induction sur le nombre de radicaux parallèles, avec l'axiome *Résidus*. \square

1.2 Des étiquettes pour un exemple informel

Notre objectif est de manipuler des systèmes abstraits de réécriture de termes dont les termes sont étiquetés et représentent des graphes. Dans un tel cadre un pas de réduction agit donc à la fois sur les étiquettes et sur la structure des termes sous-jacents. Dans cette section nous nous intéressons aux mécanismes qui permettent, à partir d'un terme source étiqueté et d'un mécanisme de réduction sur la structure non étiquetée sous-jacente, de construire des étiquettes pour le terme cible.

Pour assurer la préservation de la propriété de partage à travers laquelle les termes étiquetés peuvent représenter des graphes, nous allons introduire un ensemble d'axiomes contraignant les relations qu'entretiennent les étiquettes du terme source et du terme cible.

Dans cette section nous allons construire de toutes pièces un étiquetage pour un système de réécriture du premier ordre exemple. Les coulisses de cette construction présenteront conjointement des mécanismes d'étiquetage et les cinq axiomes du cadre axiomatique défini dans la section 1.3 (définition 1.15).

Considérons une signature $\mathcal{S} = \{a, f, g, h\}$ où les symboles a , f , g , et h ont pour arités respectives 0, 2, 1, et 2, prenons la règle de réduction

$$h(x, y) \rightarrow f(x, h(x, y))$$

et analysons la réduction de graphe suivante, où le symbole h est dupliqué mais où un partage est maintenu sur l'expression $g(a)$. Le radical et le réduit sont

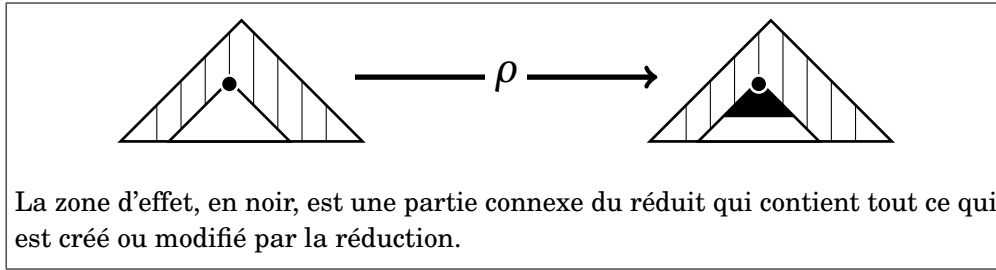
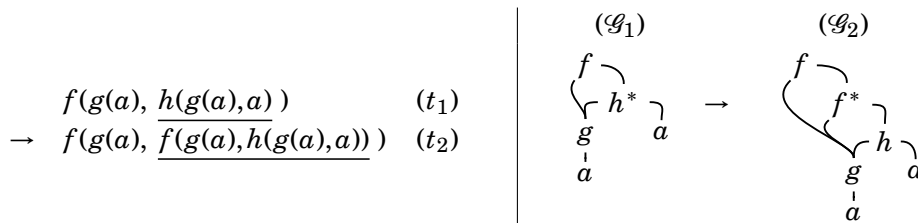


FIGURE 1.4: Zone d'effet

soulignés dans la représentation en termes, et leurs racines sont marquées d'une étoile dans la représentation en graphes.



Nous allons représenter le graphe \mathcal{G}_1 par un étiquetage t_1^l du terme t_1 dans lequel chaque étiquette représente un nœud du graphe. Nous mettrons des étiquettes sur les symboles f , g et h . En particulier, les étiquettes des deux occurrences de g doivent être égales, et les autres étiquettes doivent être différentes. Nous pouvons prendre par exemple :

$$f^\alpha(g^\beta(a), h^\omega(g^\beta(a), a)) \quad (t_1^l)$$

Remarquons en revanche qu'il n'est pas nécessaire d'étiqueter les symboles a pour indiquer leur partage : comme nous partageons des sous-termes et comme les symboles g sont partagés, les occurrences correspondantes de a sont partagées de même.

Maintenant, la réduction de t_1 va agir sur les étiquettes selon les principes suivants. Si une étiquette symbolise un emplacement en mémoire, alors la création d'étiquettes correspond à l'allocation de mémoire. Ainsi, de nouvelles étiquettes correspondent à la création ou à la duplication de nœuds du graphe source. Nous appelons *zone d'effet* d'un pas de réduction ρ et notons $\text{effz}(\rho)$ la partie de la cible de ρ où de nouvelles étiquettes apparaissent (la zone noire dans la figure 1.4). Intuitivement, une zone d'effet petite tend à indiquer que peu de duplications ont lieu.

Pour commencer, nous ne voulons pas qu'un pas de réduction modifie son contexte : la zone d'effet est contenue dans le réduit $f(g(a), h(g(a), a))$ et le contexte étiqueté $f^\alpha(g^\beta(a), \square)$ conserve ses étiquettes.

Ensuite, remarquons que certains sous-termes du réduit sont des copies conformes de certains sous-termes du radical. C'est le cas par exemple de $h(g(a), a)$ et la première occurrence de $g(a)$. Il n'est pas nécessaire de créer de nouvelles copies de ces sous-termes. Ils peuvent donc rester en dehors de

la zone d'effet et hériter des étiquettes de leur source. L'axiome *Zone d'effet* (définition 1.9) vérifie que toute autre position est dans la zone d'effet.

Finalement, seule la nouvelle occurrence de f a besoin d'une nouvelle étiquette, par exemple γ , et nous pouvons proposer l'étiquetage t_2^l suivant de t_2 pour représenter le graphe \mathcal{G}_2 .

$$f^\alpha(g^\beta(a), f^\gamma(g^\beta(a), h^\omega(g^\beta(a), a))) \quad (t_2^l)$$

À cet instant, nous devons générer une nouvelle étiquette γ d'une manière qui ne mette pas en danger la propriété de partage : nous devons notamment éviter que γ soit égale à l'étiquette d'un sous-terme différent. De préférence, nous chercherons à avoir un procédé qui construit les nouvelles étiquettes en se servant uniquement du radical et sans requérir d'informations globales sur le contexte. Ce choix fondamental dans la construction permet de définir la réduction étiquetée comme un schéma de réécriture de termes usuel.

Pour éviter les collisions fortuites, nous allons faire apparaître dans γ des informations qui sont caractéristiques du radical réduit. Ainsi, deux radicaux différents vont engendrer des étiquettes différentes, alors que deux copies du même radical pourront naturellement avoir des évolutions similaires. Dans le cadre de la métaphore sur la mémoire, une telle caractérisation d'un radical est son identité physique, c'est-à-dire son étiquette. Ce choix est possible si et seulement si la racine du radical est étiquetée, ce qui sera imposé par un axiome *Étiquette racine*. Dans notre exemple, l'étiquette ω du symbole h dans t_1^l valide ceci. Maintenant, l'étiquette γ peut être construite explicitement à partir de l'étiquette ω et d'autres informations prises dans le radical ou le réduit. Disons par exemple que f^γ est le symbole à la position ϵ du réduit d'un radical d'identité ω , et écrivons $\gamma = [\omega, \epsilon]$.

Un tel procédé fonctionne pourvu qu'une dernière condition soit vérifiée : une fois qu'un radical d'étiquette racine ω a été réduit et a créé des étiquettes de la forme $[\omega, \epsilon]$, aucun radical d'étiquette racine ω ne doit plus être créé. Ou mieux : aucune occurrence de l'étiquette ω ne doit plus être générée.

Une première précaution à court terme consiste à changer l'étiquette de h , c'est-à-dire de faire entrer h dans la zone d'effet. Une des conditions de l'axiome *Zone d'effet* impose donc que les copies de la racine du radical soient dans la zone d'effet. Remarquons aussi que la nouvelle étiquette de h , dans $[\omega, 2]$ dans l'exemple ci-dessous, doit être différente de la nouvelle étiquette $\gamma = [\omega, \epsilon]$ créée pour f . Un axiome *Partage* impose aux différentes étiquettes créées par un même pas de réduction d'être compatibles. L'étiquetage t_2^l du terme t_2 finalement obtenu pour la représentation du graphe \mathcal{G}_2 est

$$f^\alpha(g^\beta(a), f^{[\omega, \epsilon]}(g^\beta(a), h^{[\omega, 2]}(g^\beta(a), a))) \quad (t_2^l)$$

Enfin, nous devons assurer le maintien à long terme de cette propriété. Pour ceci, nous utilisons deux ingrédients :

- Un ordre partiel \leq sur les étiquettes tel que $\omega \leq [\omega, p]$ pour toute étiquette ω et toute position p . Deux axiomes *Progression* et *Héritage* contrôlent la progression des étiquettes d'un terme par rapport à cet ordre.

- Un invariant d'*indépendance* sur les termes étiquetés qui bannit la cohabitation dans un terme de deux étiquettes α et β telles que $\alpha < \beta$ (*propriété d'indépendance*, définition 1.17).

Dans ce cadre nous avons tous les ingrédients nécessaires à la preuve de la préservation de la propriété de partage par la réduction parallèle (théorème de partage 1.19). Nous pourrions donc justifier que la réduction de termes étiquetés décrit fidèlement une réduction de graphes.

1.3 Le cadre axiomatique

Cette section contient la définition des *systèmes de partage étiquetés* (*Sharing-via-Labeling Systems*, *SLS*, définition 1.15) et de leur *réduction parallèle étiquetée* (définition 1.16), ainsi que la preuve du fait que cette dernière préserve la propriété de partage (théorème 1.19).

La construction suit les principes présentés à la section 1.2 ci-dessus en deux étapes. D'abord sont définis les *ATRS marqués* (définition 1.9), qui enrichissent les *ATRS* d'un marquage des effets potentiels d'un pas de réduction (sa zone d'effet, encadrée par l'axiome *Zone d'effet*). Enfin, les *SLS* sont définis comme des *ATRS marqués* dont les termes sont partiellement étiquetés et vérifiant les quatre axiomes *Étiquette racine*, *Progression*, *Héritage* et *Partage*.

1.3.1 ATRS Marqués

Un *ATRS* peut être étendu par des *zones d'effet* identifiant dans chaque réduction une partie du réduit contenant toutes les positions susceptibles d'avoir été créées ou modifiées par la réduction (figure 1.4). L'axiome *Zone d'effet* exprime cette propriété par contraposition : tout sous-terme qui a une intersection vide avec la zone d'effet doit être syntaxiquement égal à un sous-terme strict du radical. Ceci forme un *ATRS marqué*.

Définition 1.9 (*ATRS marqué*). Soit $\Sigma = (\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ un *heptuplet* tel que :

- $(\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx})$ est un *ATRS*.
- Pour toute réduction $\rho \in \mathcal{R}$, $\text{effz}(\rho)$ est un segment initial (définition 0.25) de $\text{pos}(\text{duct}(\rho))$ appelé **zone d'effet** de ρ .

Σ est un *ATRS marqué* si l'axiome suivant est vérifié :

Zone d'effet Pour tout pas de réduction $\rho \in \mathcal{R}$ et toute position p telle que $p \in \text{pos}(\text{duct}(\rho)) \setminus \text{effz}(\rho)$, il existe une position $q \in \text{pos}(\text{dex}(\rho))$ telle que $q \neq \epsilon$ et $\text{dex}(\rho)|_q = \text{duct}(\rho)|_p$.

Remarque 1.10. La condition $q \neq \epsilon$ assure que toute copie de la racine du radical est dans la zone d'effet, ce qui va forcer la génération d'une ou plusieurs nouvelles étiquettes. Cette précaution est utile à la préservation de la propriété dite d'*indépendance* (définition 1.17, lemme 1.18) qui sera centrale à la preuve de partage (théorème 1.19).

Exemple 1.11.

Soit ρ un pas de réduction tel que

$$\begin{aligned} \text{dex}(\rho) &= g(f(a, b)) \\ \text{duct}(\rho) &= h(g(f(a, b)), a) \end{aligned}$$

Tout segment initial de $\{\epsilon, 1, 11, 111, 112, 2\}$ qui contient $\{\epsilon, 1\}$ est une zone d'effet admissible pour ρ . La colonne gauche du tableau suivant en donne des exemples, avec à droite les préfixes de $\text{duct}(\rho)$ correspondants.

$\{\epsilon, 1\}$	$h(g(\square), \square)$
$\{\epsilon, 1, 2\}$	$h(g(\square), a)$
$\{\epsilon, 1, 11, 111, 112, 2\}$	$h(g(f(a, b)), a)$

Le chapitre 3 illustrera l'impact du choix d'une zone d'effet.

Remarquons également qu'un pas de réduction ρ qui opère un écrasement (*collapsing reduction*) peut avoir une zone d'effet vide.

Exemple 1.12.

Soit ρ un pas de réduction tel que

$$\begin{aligned} \text{dex}(\rho) &= g(f(a, b)) \\ \text{duct}(\rho) &= f(a, b) \end{aligned}$$

Alors il est possible de définir

$$\text{effz}(\rho) = \emptyset$$

1.3.2 Systèmes de partage étiquetés (SLS)

Les systèmes de partage étiquetés (*Sharing-via-Labeling Systems, SLS*) sont définis comme une spécialisation des *ATRS marqués* où la signature contient des symboles potentiellement étiquetés et où les étiquettes sont partiellement ordonnées (par une relation \leq).

L'axiome *Étiquette racine* demande au symbole racine de tout radical d'être étiqueté. Les axiomes *Progression* et *Héritage* contrôlent la progression des étiquettes par rapport à l'ordre partiel \leq : la *Progression* demande que les nouvelles étiquettes soient plus grandes que l'étiquette racine et l'*Héritage* que les nouvelles étiquettes ne dominent aucune étiquette qui ne serait pas présente dans le radical. L'axiome *Partage* interdit les collisions entre les étiquettes créées simultanément dans une même zone d'effet.

Définition 1.13 (Signature étiquetée). Soit \mathcal{S} une signature, et \mathcal{L} un ensemble dénombrable dont les éléments sont appelés **étiquettes** (labels). Notons $\mathcal{S}^{\mathcal{L}} = \{f^\alpha \mid f \in \mathcal{S}, \alpha \in \mathcal{L}\}$ la **signature étiquetée** qui ajoute des étiquettes de \mathcal{L} aux symboles de \mathcal{S} en préservant leur arité. La **signature partiellement étiquetée** $\mathcal{S}[\mathcal{L}]$ est définie par $\mathcal{S}[\mathcal{L}] = \mathcal{S} \cup \mathcal{S}^{\mathcal{L}}$.

Soit t un terme sur une signature $\mathcal{S}[\mathcal{L}]$. Le terme sur \mathcal{S} obtenu en retirant à t toutes ses étiquettes est noté t^* .

Soit t un terme sur une signature $\mathcal{S}[\mathcal{L}]$, et $p \in \text{pos}(t)$. Si $t(p) = f^\alpha$ avec $f \in \mathcal{S}$ et $\alpha \in \mathcal{L}$, alors on note $\tau_p(t) = \alpha$ l'**étiquette** à la position p dans t . Sinon $\tau_p(t)$ n'est pas définie. L'expression $\tau_\epsilon(t)$ désignant l'étiquette racine de t pourra être abrégée en $\tau(t)$.

Pour tout terme t , s'il existe une position $p \in \text{pos}(t)$ telle que $\tau_p(t) = \alpha$, alors nous disons que α **apparaît** dans t . Pour tout terme t et toute étiquette α , nous notons $\text{lo}_\alpha(t)$ l'ensemble des positions des occurrences de α dans t :

$$\text{lo}_\alpha(t) = \{p \in \text{pos}(t) \mid \tau_p(t) = \alpha\}$$

Dans la suite, l'écriture de $\tau_p(t)$ supposera l'existence de cette étiquette.

Définition 1.14 (Propriété de partage). *Un terme t sur une signature $\mathcal{S}[\mathcal{L}]$ a la **propriété de partage**, ce qu'on note $\mathbb{S}(t)$ (sharing), si pour toutes positions $p, q \in \text{pos}(t)$, $\tau_p(t) = \tau_q(t)$ implique $t|_p = t|_q$. Dans ce cas, nous notons \mathcal{G}_t le **graphe** décrit par t .*

Définition 1.15 (SLS). *Soit $(\mathcal{L}, \leq, \Sigma)$ un triplet tel que :*

- \mathcal{L} est un ensemble dénombrable d'étiquettes.
- \leq est un ordre partiel sur \mathcal{L} dont nous notons $<$ l'ordre strict.
- $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué sur une signature étiquetée $\mathcal{S}[\mathcal{L}]$.

Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS si les axiomes suivants sont vérifiés :

Étiquette racine *Pour tout pas de réduction $\rho \in \mathcal{R}$ l'**étiquette racine** $\tau(\text{dex}(\rho))$ est définie. Nous la notons en abrégé $\tau(\rho)$.*

Progression *Pour tout pas de réduction $\rho \in \mathcal{R}$ et toute étiquette $\alpha \in \mathcal{L}$ apparaissant dans $\text{effz}(\rho)$, $\tau(\rho) \leq \alpha$. Si de plus $\text{effz}(\rho)$ n'est pas un singleton, alors $\tau(\rho) < \alpha$.*

Héritage *Pour tout pas de réduction $\rho \in \mathcal{R}$, toute étiquette $\beta \in \mathcal{L}$ apparaissant dans $\text{effz}(\rho)$ et toute étiquette $\alpha \in \mathcal{L} \setminus \{\tau(\rho)\}$, si $\alpha < \beta$ alors il existe une étiquette $\gamma \in \mathcal{L}$ apparaissant dans $\text{dex}(\rho)$ telle que $\alpha < \gamma$.*

Partage *Pour tout pas de réduction $\rho \in \mathcal{R}$ tel que $\mathbb{S}(\text{dex}(\rho))$ et pour toutes positions $p, q \in \text{effz}(\rho)$, si $\tau_p(\text{duct}(\rho)) = \tau_q(\text{duct}(\rho))$ alors nous avons $\text{duct}(\rho)|_p = \text{duct}(\rho)|_q$.*

*Une étiquette $\alpha \in \mathcal{L}$ est **initiale** s'il n'existe aucune étiquette $\beta \in \mathcal{L}$ telle que $\beta \leq \alpha$. Un terme t est **initial** si toutes ses étiquettes sont initiales et différentes.*

L'axiome *Progression* permet qu'une étiquette de la zone d'effet soit égale à l'étiquette racine du radical uniquement dans le cas où la zone d'effet est un singleton, c'est-à-dire uniquement dans le cas où il n'y a qu'une étiquette à créer.

Remarquons que la réduction étiquetée sans partage \rightarrow ne préserve pas la propriété de partage : seule la réduction parallèle étiquetée (définition 1.16) et les séquences de réduction correspondantes ont un sens graphique. La réduction

parallèle étiquetée est la combinaison de tous les pas de réduction portant une étiquette racine donnée, et est une réduction parallèle au sens de la définition 1.6.

La condition sur les pas de réduction équivalents dans la définition suivante assure que le radical réduit en parallèle est bien présent à chaque occurrence de l'étiquette ciblée.

Définition 1.16 (Réduction parallèle étiquetée). *Soit t un terme dans un SLS tel que $\mathbb{S}(t)$, et ω une étiquette apparaissant dans t . Par $\mathbb{S}(t)$ les positions de $1_{\circ_\omega}(t)$ sont deux à deux disjointes. Soit un pas de réduction $\rho \in \mathcal{R}$ tel que $\tau(\rho) = \omega$ et pour toute position $p \in 1_{\circ_\omega}(t)$ il existe un pas de réduction $\rho_p \in \mathcal{R}(t, p)$ équivalent à ρ . Alors t se réduit parallèlement vers $t' = t[\text{duct}(\rho)]_{1_{\circ_\omega}(t)}$, noté $t \Rightarrow_\omega t'$.*

La préservation de la propriété de partage requiert un invariant auxiliaire de la réduction parallèle étiquetée :

Définition 1.17 (Propriété d'indépendance). *Un terme t dans un SLS a la **propriété d'indépendance**, ce que nous notons $\mathbb{I}(t)$, s'il n'existe pas deux étiquettes $\alpha, \beta \in \mathcal{L}$ apparaissant dans t telles que $\alpha < \beta$.*

Cet invariant, combiné à l'axiome *Progression*, assure notamment que les étiquettes créées dans la zone d'effet d'un pas de réduction n'entrent pas en collision avec les étiquettes du contexte.

Lemme 1.18 (Indépendance). *Dans tout SLS, si $\mathbb{S}(t)$, $\mathbb{I}(t)$ et $t \Rightarrow_\omega t'$, alors $\mathbb{I}(t')$.*

Démonstration. Par définition de la réduction parallèle étiquetée, il existe un pas de réduction ρ tel que $\tau(\rho) = \omega$ et tous les sous-termes étiquetés par ω dans t sont remplacés par $\text{duct}(\rho)$ dans t' . Soient α, β deux étiquettes apparaissant dans t' telles que $\alpha < \beta$. Raisonnons par cas sur la position de β .

- I) Si β apparaît dans t , alors par $\mathbb{I}(t)$ il n'y a pas d'occurrence de α dans t . Donc α apparaît dans $\text{effz}(\rho)$, et par axiome *Progression* $\omega \leq \alpha$. Par transitivité $\omega < \beta$, ce qui contredit $\mathbb{I}(t)$.
- II) Si β n'apparaît pas dans t , alors β apparaît dans $\text{effz}(\rho)$. Raisonnons par cas sur la position de α .
 - i) Si α apparaît dans t .
 - Si $\omega = \alpha$, alors ω apparaît dans t' . Par définition de la réduction parallèle étiquetée ω apparaît dans $\text{duct}(\rho)$, et par axiome *Zone d'effet* ω apparaît dans $\text{effz}(\rho)$.
 - Si $\text{effz}(\rho)$ est un singleton alors $\omega = \beta$, donc $\alpha = \beta$, ce qui contredit $\alpha < \beta$.
 - Sinon, alors par axiome *Progression* $\omega \neq \alpha$, ce qui n'est pas.
 - Si $\omega \neq \alpha$, alors par axiome *Héritage* il existe γ apparaissant dans $\text{dex}(\rho)$ telle que $\alpha < \gamma$, ce qui contredit $\mathbb{I}(t)$.
 - ii) Si α n'apparaît pas dans t , alors α apparaît dans $\text{effz}(\rho)$.
 - Si $\omega = \alpha$.
 - Si $\text{effz}(\rho)$ est un singleton alors $\alpha = \beta$, ce qui contredit $\alpha < \beta$.
 - Sinon, alors par axiome *Progression* $\omega \neq \alpha$, ce qui n'est pas.

- Si $\omega \neq \alpha$, alors par axiome *Héritage* il existe une étiquette γ apparaissant dans $\text{dex}(\rho)$ telle que $\alpha < \gamma$, ce qui contredit $\mathbb{I}(t)$. \square

Nous avons maintenant tous les outils en main pour prouver le théorème principal de ce chapitre, qui assure que la réduction parallèle étiquetée d'un SLS préserve la propriété de partage et donc décrit une transformation de graphes.

Théorème 1.19 (Partage). *Dans tout SLS, si $\mathbb{S}(t)$, $\mathbb{I}(t)$ et $t \Rightarrow_\omega t'$, alors $\mathbb{S}(t')$.*

Démonstration. Par définition de la réduction parallèle étiquetée, il existe un pas de réduction ρ tel que $\tau(\rho) = \omega$ et tous les sous-termes étiquetés par ω dans t sont remplacés par $\text{duct}(\rho)$ dans t' . Soient $p', q' \in \text{pos}(t')$. Supposons $\tau_{p'}(t') = \tau_{q'}(t') = \alpha$. Raisonnons par cas sur p', q' .

- Si ni p' ni q' n'est dans la zone d'effet d'un radical réduit, alors par axiome *Zone d'effet* il existe une position p (resp. q) dans $\text{pos}(t)$ telle que $t(p) = t'(p')$ (resp. $t(q) = t'(q')$). En particulier $\tau_p(t) = \tau_q(t) = \alpha$, et $t|_p = t|_q$ par $\mathbb{S}(t)$. Soient $P = \{p_1 \mid \tau_{p_1}(t|_p) = \alpha\}$ et $Q = \{q_1 \mid \tau_{q_1}(t|_p) = \alpha\}$. Alors $P = Q$, et par définition de la réduction parallèle étiquetée

$$t'|_{p'} = t|_p[\text{duct}(\rho)]_P = t|_q[\text{duct}(\rho)]_Q = t'|_{q'}$$

- Si p' et q' sont toutes deux dans la zone d'effet d'un radical réduit, alors par axiome *Partage* et comme tous les réduits sont égaux à $\text{duct}(\rho)$ nous avons $t'|_{p'} = t'|_{q'}$.
- Si p' est dans la zone d'effet d'un radical réduit, et pas q' , alors d'une part $\omega \leq \alpha$ par axiome *Progression* et d'autre part α apparaît dans t par axiome *Zone d'effet*. Comme ω et α apparaissent dans t , si $\omega < \alpha$ alors nous avons une contradiction avec $\mathbb{I}(t)$. Sinon $\omega = \alpha$, ce qui implique que ω apparaît dans t' en dehors de la zone d'effet d'un radical, ce qui contredit l'axiome *Zone d'effet*. \square

Remarquons que tout terme initial t_i vérifie $\mathbb{S}(t_i)$ et $\mathbb{I}(t_i)$. Ainsi le lemme 1.18 d'indépendance et le théorème 1.19 de partage ont pour corollaire immédiat :

Corollaire 1.20 (Partage à long terme). *Dans tout SLS, si t_i est un terme initial et si une séquence de réductions parallèles étiquetées mène de t_i à un terme t , alors $\mathbb{S}(t)$.*

1.4 Exemple formel

Cette section formalise dans le cadre des SLS l'exemple de la section 1.2. Les symboles a , f , g , et h sont des constructeurs d'arité respectives 0, 2, 1, et 2. Les étiquettes sont construites à partir de positions. Définissons :

$$\begin{array}{lll} \text{Étiquettes } (\mathcal{L}) : & \alpha ::= p \mid [\alpha, p] & p \text{ position} \\ \text{Termes } (\mathcal{T}) : & t ::= x \mid a \mid F^\alpha(t_1, \dots, t_n) & F \in \{f, g, h\} \end{array}$$

La relation \leq est le plus petit ordre partiel vérifiant $\omega \leq [\omega, p]$ pour toute étiquette $\omega \in \mathcal{L}$ et toute position p .

Définissons la règle de réduction étiquetée

$$h^\omega(x, y) \rightarrow f^{[\omega, \epsilon]}(x, h^{[\omega, 2]}(x, y))$$

Un pas de réduction est l'application de cette règle dans un contexte quelconque.

Alors la réduction parallèle étiquetée préserve la propriété de partage, ce que nous allons vérifier en exprimant ce système comme un *SLS*.

Instanciation 1.21. *Considérons le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- la signature $\mathcal{S} = \{a, f, g, h\}$,
- l'ensemble \mathcal{T} des termes étiquetés selon la grammaire précédente, et
- l'ensemble \mathcal{R} des pas de réduction étiquetés.

Pour tout pas de réduction $\rho \in \mathcal{R}$, notons $\rho : c[h^\omega(t_x, t_y)] \rightarrow c[f^{[\omega, \epsilon]}(t_x, h^{[\omega, 2]}(t_x, t_y))]$ et définissons :

- $\text{dex}(\rho) = h^\omega(t_x, t_y)$
- $\text{duct}(\rho) = f^{[\omega, \epsilon]}(t_x, h^{[\omega, 2]}(t_x, t_y))$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \{\epsilon, 2\}$

Lemme 1.22. *Le système $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration.

Source & Cible \mathcal{T} contient tous les termes syntaxiques de la grammaire.

Résidus La relation de réduction est close sous tout contexte.

Zone d'effet Toute position hors de la zone d'effet est une position d'un sous-terme de t_x ou de t_y . □

Lemme 1.23. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine Toutes les occurrences de h sont étiquetées.

Progression Soit un pas de réduction $\rho \in \mathcal{R}$ d'étiquette racine ω . Toute étiquette de la zone d'effet a la forme $[\omega, \epsilon]$ ou $[\omega, 2]$, avec $\omega < [\omega, \epsilon]$ et $\omega < [\omega, 2]$.

Héritage Soit un pas de réduction $\rho \in \mathcal{R}$ d'étiquette racine ω . Soit β une étiquette de la zone d'effet. Alors β a la forme $[\omega, \epsilon]$ ou $[\omega, 2]$. En particulier tout $\alpha \in \mathcal{L}$ tel que $\alpha < \beta$ vérifie $\alpha \leq \omega$.

Partage Les deux étiquettes $[\omega, \epsilon]$ et $[\omega, 2]$ de la zone d'effet sont différentes, il n'y a donc rien à vérifier. □

Ainsi, notre système étiqueté est bien un *SLS*, et par théorème 1.19 ses étiquettes décrivent bien un système de réduction de graphes.

1.5 Comparaison d'étiquetages

Un *ATRS* Σ étant donné, il existe potentiellement de nombreuses manières de construire un *SLS* Σ^l en ajoutant des étiquettes à Σ , qui peuvent générer ou non des notions de réduction parallèle étiquetée équivalentes. À plusieurs reprises dans cette thèse (chapitres 3, 4, et 7) nous aurons ainsi à comparer différents étiquetages d'un même système de réécriture pour démontrer la bisimilarité des réductions partagées qu'ils décrivent.

La présente section introduit des outils pour permettre cette comparaison, en caractérisant en particulier les termes qui se correspondent dans différents étiquetages d'un même système sous-jacent.

Pour cela nous commençons par formaliser les critères qui permettent d'affirmer qu'un *SLS* Σ^l est un *étiquetage* d'un *ATRS* Σ . Cette caractérisation est directement issue d'une caractérisation abstraite de l'étiquetage d'un *ARS* proposée par van Oostrom [Ter03, Chap. 8]. L'idée de cette relation d'étiquetage, appliquée à un *SLS* Σ^l et un *ATRS* sous-jacent $\Sigma = (\Sigma^l)^\bullet$ tient dans les faits suivants :

- Le passage d'un terme de Σ^l à un terme de Σ est fait en retirant les étiquettes.
- Une fonction des termes de Σ dans ceux de Σ^l définit un étiquetage qualifié d'*initial*.
- Tout pas de réduction étiquetée de Σ^l se projette sur un pas de réduction non-étiquetée de Σ .
- Tout pas de réduction non étiquetée de Σ a un unique équivalent étiqueté dans Σ^l pour chaque étiquetage de sa source.

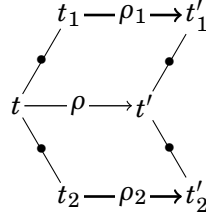
Formellement, cette relation est une notion enrichie de bisimulation. Ci-dessous, nous adaptions au cadre des *ATRS* la définition proposée par van Oostrom pour les *ARS* [Ter03, Chap. 8].

Définition 1.24 (Étiquetage d'un *ATRS*). *Soit* $(\mathcal{L}, \leq, \Sigma_l)$ *un SLS avec* $\Sigma_l = (\mathcal{S}_l[\mathcal{L}], \mathcal{T}_l, \mathcal{R}_l, \text{dex}_l, \text{duct}_l, \text{ctx}_l, \text{effz}_l)$. *Soit* $\Sigma = (\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx})$ *un ATRS. Nous dirons que* Σ_l *est un étiquetage du système* Σ *si* $\mathcal{S}_l = \mathcal{S}$, $\mathcal{T}_l^\bullet = \mathcal{T}$, *et s'il existe une fonction d'étiquetage initial* $i(\cdot)$ *de* \mathcal{T} *vers* \mathcal{T}_l *et une fonction de projection* $(\cdot)^\bullet$ *de* \mathcal{R}_l *vers* \mathcal{R} *telle que pour tout pas de réduction* $\rho_l \in \mathcal{R}_l$:

- $\text{dex}(\rho_l^\bullet) = \text{dex}_l(\rho_l)^\bullet$
- $\text{duct}(\rho_l^\bullet) = \text{duct}_l(\rho_l)^\bullet$
- $\text{ctx}(\rho_l^\bullet) = \text{ctx}_l(\rho_l)^\bullet$

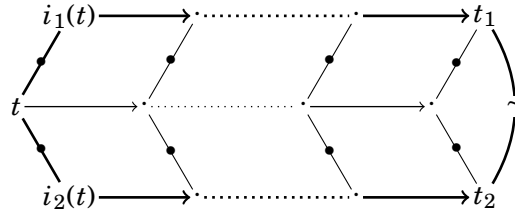
et pour tout pas de réduction $\rho \in \mathcal{R}$ *et tout étiquetage autorisé* $t_l \in \mathcal{T}_l$ *de la source* $t = \text{src}(\rho)$ *il existe un unique pas de réduction* $\rho_l \in \mathcal{R}_l$ *tel que* $\rho_l^\bullet = \rho$ *et* $\text{src}(\rho_l) = t_l$.

Cette bisimulation nous permettra en particulier dans cette thèse de comparer les séquences de réductions dans différents étiquetages d'un même *ATRS* sous-jacent. Pour ce faire, la notion de *réduction miroir* caractérise les équivalents étiquetés d'un même pas de réduction non étiqueté dans deux systèmes étiquetés différents. La définition suit ce diagramme avec les pas $\rho : t \rightarrow t'$, $\rho_1 : t_1 \rightarrow t'_1$, et $\rho_2 : t_2 \rightarrow t'_2$ tels que $\rho_1^\bullet = \rho = \rho_2^\bullet$.



Définition 1.25 (Réductions miroirs). Soient deux SLS Σ_1 et Σ_2 qui sont deux étiquetages d'un même ATRS Σ . Un pas de **réduction miroir** dans les systèmes Σ_1 et Σ_2 est une paire $(\rho_1, \rho_2) \in \mathcal{R}_1 \times \mathcal{R}_2$ telle que $\rho_1^\bullet = \rho_2^\bullet$. Nous appellerons **source** (resp. **cible**) de (ρ_1, ρ_2) la paire $(\text{src}_1(\rho_1), \text{src}_2(\rho_2))$ (resp. $(\text{tgt}_1(\rho_1), \text{tgt}_2(\rho_2))$). Cette notion s'étend aux séquences de réduction ainsi qu'à la réduction parallèle étiquetée.

De cette relation entre des séquences de réductions nous déduisons une relation entre des termes, caractérisant des termes étiquetés de deux systèmes différents pouvant être obtenus d'une même source par des séquences de réduction miroirs.



Définition 1.26 (Termes miroirs). Soient deux SLS Σ_1 et Σ_2 qui sont deux étiquetages d'un même ATRS Σ . Soient t_1 et t_2 deux termes étiquetés respectivement dans \mathcal{T}_1 et \mathcal{T}_2 . Les termes t_1 et t_2 sont **miroirs**, noté $t_1 \sim t_2$, s'il existe un terme non étiqueté $t \in \mathcal{T}$ et une séquence de réduction miroir (ρ_1, ρ_2) de source $(i_1(t), i_2(t))$ et de cible (t_1, t_2) . Nous noterons $t_1 \approx t_2$ si de plus une séquence miroir de réductions parallèles étiquetées témoigne de $t_1 \sim t_2$.

Lemme 1.27 (Termes miroirs). Soient deux SLS Σ_1 et Σ_2 qui sont deux étiquetages d'un même ATRS Σ . Soient t_1 et t_2 deux termes étiquetés tels que $t_1 \sim t_2$. Alors $t_1^\bullet = t_2^\bullet$.

Démonstration. Par récurrence sur la longueur de la séquence de réduction miroir. \square

Par définition même, la relation “termes miroirs” est une bisimulation entre les deux systèmes concernés. Un enjeu de certains chapitres ultérieurs (chapitres 3, 4, et 7) sera de prouver que cette bisimulation peut dans certains cas valoir non seulement pour la réduction simple mais aussi pour la réduction parallèle étiquetée.

1.6 Discussion

Pour conclure ce chapitre la présente section met en perspective le cadre axiomatique qui vient d'être défini. La section 1.6.1 discute des points clés de l'expressivité des *SLS*, puis la section 1.6.2 compare notre cadre avec d'autres formalisations de la réduction de graphes ou de la réduction partagée. La section 1.6.3 enfin détaille les usages qui peuvent être faits des *SLS* et les points susceptibles de receler des difficultés ou absences de difficultés notables selon les usages.

1.6.1 Expressivité du cadre axiomatique

Le système présenté dans ce chapitre axiomatise le partage par étiquettes selon trois directions : les systèmes de termes sous-jacents, le partage, et les étiquetages. Chacun de ces niveaux d'abstraction apporte son propre pouvoir d'expression. Nous les passons ici en revue dans l'ordre inverse.

Les étiquettes jouent deux rôles. Comme annoncé elles décrivent d'abord le partage, mais elles doivent aussi contenir suffisamment d'information pour permettre la génération d'étiquettes fraîches. Les axiomes *Progression* et *Héritage* utilisés à cet effet sont strictement moins restrictifs que leurs équivalents dans les approches précédentes du partage par étiquettes [Mar91, BLM07, Bal10a, Bal12b]. En particulier nos étiquettes peuvent, mais ne sont pas obligées de, contenir autant d'information causale que leurs prédécessrices. Une conséquence est que les séquences d'étiquettes et les nœuds factices qui peuvent créer des chaînes d'indirection arbitrairement longues dans de précédents travaux [Bal12b] ne sont plus nécessaires. Ceci est rendu possible par un invariant d'*indépendance* qui est strictement plus général que ses prédécesseurs comme la *maximalité* [BLM07]. Ce point est une clé de la contribution technique de ce chapitre, dont nous apprécierons la portée en particulier aux chapitres 8, 9, et 11.

Les systèmes de réduction de graphes pouvant être décrits dans le cadre axiomatique viennent de la combinaison de deux paramètres : d'abord la zone d'effet spécifie les positions pour lesquelles de nouveaux nœuds sont créés, puis l'étiquetage détermine les positions de la zone d'effet qui sont partagées ou non. Le premier paramètre est encadré par l'axiome *Zone d'effet* et le deuxième par l'axiome *Partage*, qui sont tous deux exprimés en termes d'égalité syntaxique de sous-termes. Ceci nous permet de décrire une grande variété de systèmes de réduction de graphes, dont notamment les nombreuses variantes de l'appel par nécessité, de la réduction paresseuse [Lau93, AFM⁺95, DMMZ10], et de la réduction pleinement paresseuse [Wad71, AF97, Ses97, SW10, BLM07] qui ont été étudiées sur le λ -calcul ou des calculs similaires. Certaines de ces variantes ont été étudiées dans une version précédente du partage par étiquettes [Bal12b], mais cet encodage se faisait au prix de la définition de stratégies de réduction finement ciselées pour l'adaptation à chaque cas. Nous pouvons maintenant poursuivre ces études dans notre nouveau cadre sans ce besoin de trafiquer au préalable la dynamique du système.

Enfin, le cadre abstrait pour la réécriture de termes ne fait aucune hypothèse sur la forme des règles de réduction ni sur les systèmes de termes sous-jacents, et fait peu d'hypothèses sur les conditions d'application de ces règles. La seule restriction est celle qui interdit à des pas de réduction affectant des sous-termes disjoints d'une même source d'interférer l'un avec l'autre (figure 1.3 et axiome *Résidus*). De plus, la spécification arbitraire d'un ensemble de termes permet naturellement de travailler avec des termes vérifiant certains invariants. Ainsi, tout système raisonnable de réécriture de termes est susceptible d'être représenté. Voici à titre d'exemples quelques classes de systèmes d'ordre supérieur qui sont intéressantes d'un point de vue de la théorie de la réécriture, et qui sont de plus pertinentes pour l'étude de la programmation fonctionnelle.

Calculs conditionnels Les *stratégies de réduction closes* [FMS05] sont des familles de fragments du λ -calcul adaptés à l'implémentation puisqu'ils ne nécessitent pas d' α -conversion. Ces stratégies introduisent des conditions sur la β -réduction portant sur les variables libres des radicaux. De nombreux chapitres de cette thèse concernent des systèmes en réduction faible, qui en sont un cas particulier.

Systèmes non-linéaires-droits Plusieurs exemples assez populaires de systèmes d'ordre supérieur sont basés sur des règles dont les membres droits ne sont pas linéaires [KvOvR93, Ter03]. C'est le cas en particulier de la règle de point fixe $\mu x.t \rightarrow t^{x:=\mu x.t}$ ou des fonctions `fold` et `map`.

Systèmes non-orthogonaux La réécriture d'ordre supérieur est utile à l'étude des *transformations de programmes*. Des systèmes non-orthogonaux peuvent être observés dans ce cas, comme le λ -lifting pleinement paresseux [Hug82, PJ87] que nous verrons aux chapitres 8 et 9. Un autre réservoir bien connu de systèmes non-orthogonaux est l'importante littérature sur les calculs avec *substitutions explicites* [ACCL90, Kes07]. Les substitutions explicites sont un outil important pour réduire l'écart entre la définition formelle de la β -réduction et ses diverses implémentations.

Systèmes non-séquentiels Définir des fonctions par cas au moyen d'opérations de *filtrage* est un trait très appréciable de la programmation fonctionnelle. Une difficulté théorique majeure est la gestion des erreurs de filtrage, qui sont non-séquentielles en général [JK09] et donc similaires à l'opérateur *ou parallèle*. Voir le chapitre 6.

1.6.2 Comparaison avec d'autres techniques

Le partage et la réécriture de graphes ont déjà été décrits et étudiés de nombreuses manières différentes, dont nous passons en revue les principales ici. Nous les comparons avec notre cadre axiomatique selon trois axes :

Expressivité : la quantité de systèmes de réécriture compatibles et leurs caractéristiques.

Pouvoir de partage : le niveau de partage atteignable sur les systèmes compatibles.

Complexité du formalisme : la simplicité ou complexité de la définition et de la manipulation d'un système.

Systèmes de réécriture adressés (*Adressed TRS*) Dougherty et al. [DLLL05] proposent des systèmes du premier ordre étiquetés présentant un pouvoir de partage similaire à celui des *SLS*. Ils autorisent les systèmes non-orthogonaux, mais aussi des cycles représentant des termes infinis, ce que nous ne faisons pas ici. Ils ne font aucune mention de l'ordre supérieur. Leur technologie est légèrement plus complexe que la réécriture de termes : les nouvelles étiquettes sont choisies globalement fraîches. Par conséquent la réduction d'un radical dépend de son contexte global et la réduction parallèle ne se décompose pas en séquences de pas de réduction simples.

Graphes-termes (*Termgraphs*) Les graphes-termes [BvEG⁺87, Plu99, Blo01, Kah98] apportent des mécanismes de partage pour tout système du premier ordre ou d'ordre supérieur, avec un pouvoir de partage équivalent à celui des *SLS* dans de nombreux cas, et autorisent également les cycles [BBCK07]. Leur formalisme est bien plus complexe que celui des *SLS*, car les graphes-termes requièrent des critères de correction subtils, en particulier à l'ordre supérieur, et des notions non triviales de morphismes de graphes avec lieux.

Appel par nécessité (*Call-by-Need*) Le λ -calcul en appel par nécessité [AFM⁺95, AF97, MOW98, CF12] et sa version pleinement paresseuse [AF97] donnent une traduction simple et élégante de l'algorithme de graphes de Wadsworth pour l'évaluation du λ -calcul [Wad71]. Nobuko Yoshida [Yos94] développe également des idées similaires en utilisant des substitutions explicites à la place de constructions let. Ces travaux réussissent à n'utiliser que des termes et de la réduction simple de termes. Cependant, ils nécessitent des règles structurelles ou des équations structurelles. Notons que ce dernier point a été amélioré récemment [CF12]. Concernant l'expressivité, ces travaux traitent du cas particulier du λ -calcul, qui est un système orthogonal.

Réseaux d'interaction Les réseaux d'interaction permettent d'implémenter de vastes classes de systèmes de réécriture du premier ordre et d'ordre supérieur [FM98, FMSW09, Mac98, Mac04]. Ils sont connus en particulier pour la réalisation de partages non-optimaux mais néanmoins très puissants [Mac98, Mac04]. La limite de leur expressivité tient à leur nature séquentielle. De plus, la décomposition de chaque opération qu'ils imposent rend le lien d'un système d'interaction avec sa spécification sous forme de termes techniquement délicat.

Partage optimal (*Sharing graphs*) Les implémentations optimales de systèmes d'ordre supérieur [Lé80, Kat90, Lam90, AL96, AG98] mettent en œuvre un partage plus approfondi que ce que nous pouvons faire ici. En particulier, les *sharing graphs* permettent le partage d'un fragment de terme avec certains fragments de ses sous-termes, ce qui forme des cycles qui ne peuvent pas

exister dans les *SLS*. En outre ces implémentations s'appliquent à travers les systèmes d'interaction [AL96] à une large classe de systèmes d'ordre supérieur orthogonaux. Cependant, ces travaux ne s'appliquent pas à tous les systèmes orthogonaux, et ne s'appliquent pas du tout aux systèmes non-orthogonaux. Enfin, si ces graphes n'utilisent que des règles de réduction locales et très simples, ils nécessitent aussi des structures de contrôle additionnelles au comportement subtil qui, comme dans les réseaux d'interaction, rendent complexe le lien avec les termes usuels.

Versión préliminaire [Bal12a] La présentation des *SLS* faite dans ce chapitre a quelques différences mineures avec l'article [Bal12a].

- L'axiome *Séparation* a été renommé en *Partage*.
- L'axiome *Contribution* est séparé en les deux axiomes *Progression* et *Héritage*.
- La relation transitive et irréflexive \hookrightarrow est remplacée par un ordre \leq .
- Il est maintenant permis, uniquement au cas où la zone d'effet est un singleton, que l'étiquette racine d'un radical apparaisse dans un radical. Ce point sera utile aux chapitres 8 et 9.

1.6.3 Utilisabilité du cadre axiomatique

Les travaux axiomatiques ont en général ceci d'élégant qu'ils capturent des propriétés jugées intéressantes dans un cadre significativement épuré. Le principe est d'identifier des conditions générales (les axiomes) permettant l'établissement d'un certain résultat ou d'une certaine théorie. Un théorème axiomatique peut donc être vu comme un *méta-théorème*, duquel nous pouvons déduire des théorèmes concrets à chaque fois que nous sommes capables d'exhiber un système vérifiant les *méta-hypothèses* du méta-théorème, c'est-à-dire en particulier les axiomes.

L'utilité d'un résultat axiomatique est donc la même que celle de n'importe quel théorème mathématique : il est muet si ses hypothèses ne sont pas acquises. La valeur d'un tel résultat n'est donc pas seulement ce qu'il produit, mais aussi le chemin parcouru entre les hypothèses et la conclusion. Un résultat axiomatique est donc pleinement appréciable quand il repose sur des structures simples à construire et des axiomes faciles à vérifier.

Analysons donc les structures et les axiomes utilisés ici, puisqu'ils sont les conditions d'application des résultats de ce chapitre.

Représentation des réductions. La présentation des réductions sous la forme radical/réduit/contexte (*dex/duct/ctx*) est en prise directe avec les pratiques usuelles en réécriture de termes. Il n'y a donc pas de transformation nécessaire ici.

Source & Cible et invariants. Il y a deux cas possibles ici. Soit tous les termes étiquetés sont acceptés et cet axiome est transparent (chapitres 4 et 7 par exemple), soit nous avons besoin d'invariants additionnels sur la réduction des termes étiquetés, et la vérification de cet axiome est

alors ramenée à la préservation de ces invariants (chapitres 2, 3, 8 et 9 notamment).

Résidus. Cet axiome est évident quand la réduction est close par tout contexte ou soumise à des conditions locales (c'est-à-dire concernant le sous-terme ou les positions préfixes). C'est le cas de tous les cadres de réécriture de termes conditionnelle que je connaisse.

Zone d'effet. Il y a deux manières d'approcher la définition de la zone d'effet, en fonction de l'utilisation faite de l'axiomatique. Si nous avons déjà construit un système étiqueté et voulons simplement vérifier qu'il décrit un partage, alors la zone d'effet est imposée et recouvre exactement la région dans laquelle les étiquettes sont changées. Si en revanche nous nous servons de l'axiomatique comme guide dans la construction d'un étiquetage pour un système non étiqueté donné, alors la définition de la zone d'effet est plutôt souple : n'importe quelle région contenant les régions créées ou modifiées par la réduction peut faire l'affaire. En particulier l'axiome est trivialement vérifié si la zone d'effet couvre tout le réduit. Cette solution facile étant peu satisfaisante quant à la qualité du partage produit, il est préférable d'opter pour une zone "minimale". Nous obtiendrons dans cette thèse de telles zones en considérant les membres droits des règles de réduction et en isolant les parties neuves ou potentiellement soumises à une substitution.

Étiquette racine. Cette axiome se vérifie à la définition des termes étiquetés.

Ordre sur les étiquettes, Progression et Héritage. À aucun moment dans cette thèse ces axiomes ne seront une difficulté, car nous répéterons le schéma suivant : prendre des étiquettes librement générées par une grammaire donnée, définir un ordre inclus dans l'ordre "sous-étiquette" sur cette grammaire, et construire les étiquettes de la zone d'effet en utilisant uniquement les étiquettes du radical et des étiquettes initiales. Notons néanmoins que si nous prenons un système d'étiquettes qui a été généré différemment il y a ici plus de travail, puisqu'il faut d'abord "deviner" un ordre qui fera l'affaire, puis vérifier les axiomes. Comme l'axiome *Héritage* a tendance à contrôler drastiquement la comparabilité des étiquettes, une stratégie de divination qui semble raisonnable consiste à partir de la relation vide, puis ajouter ce qui est imposé par l'axiome *Progression* en vérifiant que cela n'entre pas en conflit avec l'axiome *Héritage*. Estimer la complexité de ce processus est malaisé du fait de la rareté des cas concrets à étudier, mais nous verrons que les étiquetages issus de la littérature sur le partage optimal ne posent pas de difficulté.

Partage. Dans le cas où toutes les étiquettes de la zone d'effet sont différentes, il n'y a rien à vérifier pour cet axiome. Ce sera le cas par exemple aux chapitres 2, 4, 5, 6, 7, et 10. Remarquons que ce cas apparaît aussi à "l'état naturel" avec les étiquetages utilisés dans l'étude du partage optimal [Mar91, Mar92, BLM07]. Les chapitres 8 et 9 en montreront un cas particulier plus simple encore dans lequel certaines règles de réduction génèrent systématiquement une zone d'effet qui est un singleton. Sinon, la

difficulté de la vérification varie à chaque cas. Aux chapitres 3 et 11, nous construirons les étiquettes de la zone d'effet en combinant les étiquettes du radical, et le tout sera de s'assurer que chaque nouvelle étiquette reflète bien la modification (par exemple la substitution) qui a affecté la position correspondante. Si difficultés il y a, elles seront généralement dues au non respect de la convention de Barendregt, point qui sera discuté le moment venu.

Il y a parfois un équilibre intéressant à trouver entre plusieurs de ces points. Notamment, une zone d'effet plus petite pourra être plus délicate à définir, mais en même temps apporter une grande simplification à la vérification de l'axiome de partage, et ce jusqu'à sa trivialisaiton (voir chapitres 4 et 10). Nous verrons aussi aux chapitres 2 et 8 comment les difficultés potentielles de l'axiome de partage peuvent encore être annulées par la combinaison d'une zone d'effet restreinte et d'invariants sur les termes étiquetés.

Nous pouvons donc tirer un bilan sur les deux utilisations possibles de ce cadre axiomatique, qui présentent des difficultés et des facilités différentes.

- L'utilisation pour l'étude d'un système de réécriture étiqueté pré-existant peut demander de l'imagination à deux endroits : la définition d'invariants sur les termes étiquetés et la construction d'un ordre sur les étiquettes. Notons toutefois que le second point est guidé en partie. Le reste des définitions est directement déduit du système concret observé. La vérification de l'axiome de partage dépend ensuite grandement des invariants qui auront été introduits, et la vérification des axiomes de progression et d'héritage peut demander du travail.
- L'utilisation pour la création d'un système étiqueté simulant un système de réduction de graphes peut également demander de l'imagination à deux endroits : une fois encore au niveau de la définition d'invariants sur les termes étiquetés, mais également au niveau de la définition de la zone d'effet. L'ordre sur les étiquettes en revanche et les axiomes associés sont immédiats en suivant les constructions illustrées dans cette thèse. La preuve de l'axiome de partage peut demander du travail, en fonction de la zone d'effet et des invariants établis.

Bilan du chapitre

Dans ce chapitre nous avons tout d'abord vu un cadre abstrait pour la réécriture de termes (*ATRS*, définition 1.3), dans lequel peuvent être exprimés des pas de réduction simple aussi bien que des étapes de réduction parallèle. Nous avons ensuite construit dans ce cadre un système axiomatique (*SLS*, définition 1.15) caractérisant des termes étiquetés munis d'une notion de réduction parallèle qui simule des étapes de réduction de graphes.

La clé de cette représentation d'un système de réécriture de graphes par un système de réécriture de termes étiquetés est la préservation de la propriété de partage (théorème 1.19) qui assure que deux sous-termes portant la même

étiquette sont syntaxiquement égaux et peuvent donc être partagés en toute sécurité.

Enfin, nous avons décrit les différents étiquetages d'un *ATRS* donné (définition 1.24), et caractérisé les termes étiquetés qui se correspondent dans différents étiquetages d'un même *ATRS* (termes miroirs, définition 1.26).

Le formalisme des *SLS* sera utilisé dans l'ensemble de cette thèse pour la représentation de la réduction partagée dans de nombreux systèmes de réécriture de termes d'ordre supérieur, y compris des systèmes non-orthogonaux (chapitres 4, 6, 7, 8, et 9 en particulier).

La notion de termes miroirs sera quant à elle utilisée pour comparer différents systèmes de réduction partagée générés par des *SLS* différents, notamment au chapitre 3 pour montrer l'unité de différentes approches de la pleine paresse trouvées dans la littérature et au chapitre 4 pour vérifier la correction de la généralisation à la réécriture d'ordre supérieur.

DEUXIÈME PARTIE

PLEINE PARESSE

Introduction

La présente partie a un double objectif : d'abord caractériser la notion de partage pleinement paresseux telle qu'elle peut être trouvée dans les littératures sur les langages de programmation fonctionnelle et sur le λ -calcul, puis généraliser cette notion à un cadre de réécriture d'ordre supérieur.

D'un bout à l'autre cette exploration s'appuiera sur la représentation du partage par des termes étiquetés dans le cadre axiomatique de la première partie.

Le chapitre 2 présente la source première de l'appel par nécessité et de la pleine paresse à travers la technique de réduction de graphes proposée par Wadsworth pour l'évaluation du λ -calcul [Wad71]. Ceci sera motif à la définition d'un premier étiquetage du λ -calcul.

Le chapitre 3 passe ensuite en revue d'autres définitions qui ont été données de la pleine paresse pour le λ -calcul et des systèmes assimilés [AF97, Ses97, SW10, BLM07]. Une présentation uniforme de ces différentes approches est obtenue au moyen d'un étiquetage paramétrique du λ -calcul, dont l'étiquetage du chapitre 2 est un cas particulier. Cette représentation uniforme permet une comparaison formelle des différents systèmes et en particulier la preuve de leur équivalence.

Le chapitre 4 enfin, fort de l'analyse menée dans les deux chapitres précédents, propose une généralisation à la réécriture d'ordre supérieur de la notion de partage pleinement paresseux, en sélectionnant parmi les idées des différentes approches vues au chapitre 3 celles qui semblent les plus robustes.

C'est sur la notion de pleine paresse obtenue au terme de cette partie que se poursuivront les investigations de cette thèse.

CHAPITRE 2

L'algorithme de Wadsworth

Sommaire

2.1 Les λ-graphes sur un exemple	84
2.2 Structure du λ-calcul faible	86
2.3 Représentation étiquetée des λ-graphes	90
2.4 Discussion : appel par nécessité	94

Dans l'ultime chapitre de sa thèse, Wadsworth [Wad71] a proposé la première technique d'évaluation pour le λ -calcul combinant le pouvoir de normalisation de l'appel par nom avec un contrôle des duplications aussi bon, et même meilleur, que celui opéré par l'appel par valeur. Il a baptisé ce mécanisme « appel par nécessité ».

La réalisation de l'appel par nécessité combine deux idées. Il s'agit d'abord de ne réduire que des radicaux *nécessaires*, ce qui peut être fait en λ -calcul avec l'appel par nom, qui réduit toujours le β -radical le plus à gauche dans l'écriture du terme. Ainsi, si le terme considéré admet une forme normale alors tout pas de réduction suivant cette stratégie est un progrès vers cette forme normale. Ensuite la duplication des termes inhérente au λ -calcul est limitée par des mécanismes de *mémoïsation*, qui permettent de réutiliser les résultats de calculs précédents.

La mémoïsation réalise l'effet suivant. Considérons un argument a qui est dupliqué par une β -réduction. Nous obtenons deux copies identiques a_1 et a_2 . Au moment où pour la première fois la valeur d'une de ces deux copies, disons a_1 , est requise, nous effectuons l'évaluation de a_1 et obtenons une certaine valeur v_1 . Ensuite, le jour où la valeur de la deuxième copie a_2 sera elle aussi requise, la valeur v_1 pourra être immédiatement utilisée, sans effectuer de nouveau calcul sur a_2 .

Cet effet peut être réalisé concrètement à l'aide de manipulations explicites d'une mémoire. Par exemple l'argument a qui est dupliqué dans le λ -calcul est plutôt stocké en mémoire à une adresse l , et deux pointeurs font référence à cet emplacement mémoire l . Alors, au moment où la valeur de la première copie est requise, le résultat v de l'évaluation de a remplace a à l'adresse l en

mémoire. Ainsi il n'y aura effectivement plus aucun calcul à faire lorsque nous rechercherons l'argument a à l'adresse l en mémoire. C'est ce mécanisme qui a été mis en place dès les premières mises en œuvre de langages de programmation à évaluation paresseuse [FW76, HJ76] et qui sert d'idée directrice à la sémantique de la paresse de Launchbury [Lau93].

Wadsworth donne une vue abstraite de ce comportement en évitant toute mention explicite à la mémoire. Il a introduit pour ceci les λ -graphes qui représentent des λ -termes avec du partage. Ce modèle de graphes, en tant que représentation abstraite de la mémoïsation, ne dit pas comment la réaliser en pratique. En revanche les graphes permettent de décrire le comportement de la réduction partagée. Ainsi, plus que les graphes eux-mêmes ce sont leurs séquences de réduction qui nous intéressent : nous ne supposons pas que les graphes contiennent des informations utilisables sur la manière dont est structurée la mémoire, car nous n'observerons des graphes que les espaces de réduction qu'ils définissent et les longueurs des séquences de réduction associées.

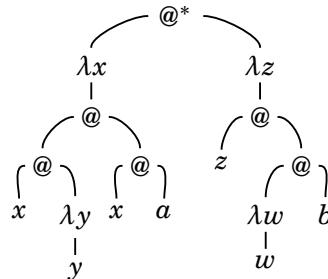
L'objectif de ce chapitre est l'expression de la technique de Wadsworth dans un système de partage étiqueté. Nous verrons d'abord en section 2.1 la technique de Wadsworth en œuvre sur un exemple. La section 2.2 donne quelques compléments sur le λ -calcul et établit un lemme fondamental de la β -réduction faible. Nous exprimerons ensuite en section 2.3 la technique de Wadsworth sous la forme d'un λ -calcul étiqueté qui est un *SLS*. La section 2.4 conclut ce chapitre par des discussions sur les stratégies de réduction et sur la représentation graphique des variables liées.

2.1 Les λ -graphes sur un exemple

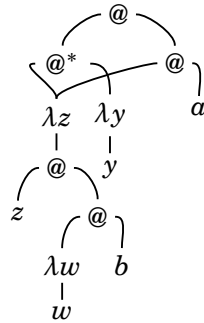
Cette section présente un exemple de réduction de graphes selon la technique de Wadsworth. Considérons le λ -terme :

$$t_0 = (\lambda x.(x\lambda y.y)(xa))\lambda z.z((\lambda w.w)b)$$

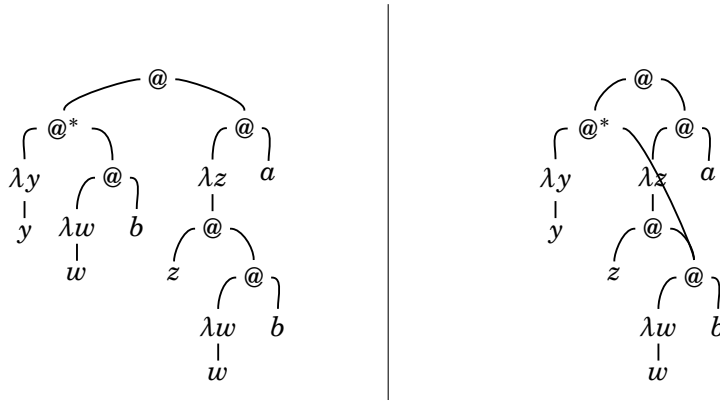
Le terme t_0 est représenté par le graphe \mathcal{G}_0 dessiné ci-dessous. Un nœud @ représente une application, et l'étoile désigne la position du prochain radical réduit.



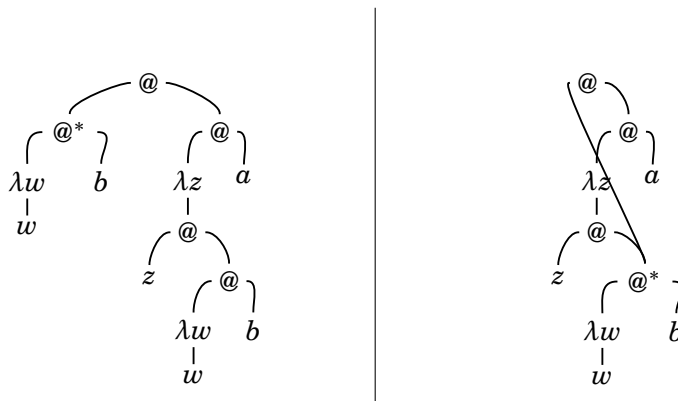
La fonction du radical utilisant deux fois la variable x , du partage apparaît dans le graphe \mathcal{G}_1 pour éviter la duplication de l'argument $\lambda z.z((\lambda w.w)b)$.



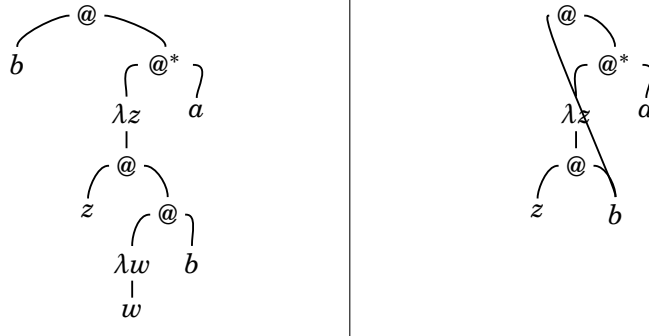
L'étape suivante implique l'instanciation d'une fonction $\lambda z.z((\lambda w.w)b)$ qui est partagée. La gestion naïve du partage duplique alors l'intégralité de cette fonction (à gauche ci-dessous). Une gestion plus avancée ne duplique que ce qui dépend de z et évite la duplication de l'expression libre $(\lambda w.w)b$ (à droite).



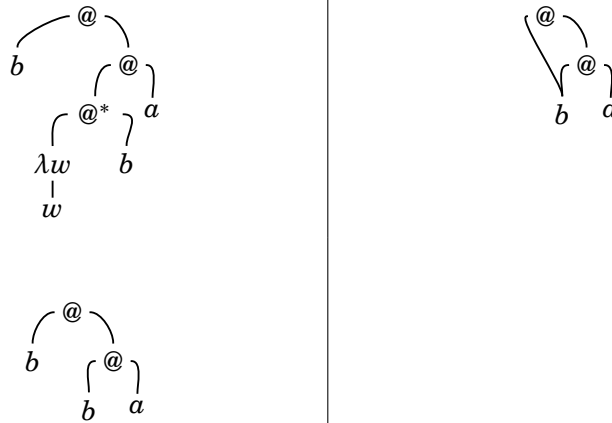
La λ -abstraction $\lambda y.y$ est maintenant appliquée à un argument $(\lambda w.w)b$ éventuellement partagé dans le graphe. Ceci ne modifie en rien ledit argument et ne nécessite aucune duplication. Le sous-terme $(\lambda w.w)b$ reste partagé à droite, et séparé à gauche.



Jusqu'ici, le partage n'a servi qu'à rendre le terme plus compact et n'a pas eu d'impact sur la réduction proprement dite. Pour la prochaine étape en revanche le radical $(\lambda w.w)b$ à réduire est partagé dans le cas de droite, et une seule étape dans le graphe va correspondre à deux étapes dans le terme sous-jacent.



Dans le cas de droite il ne reste alors plus qu'une étape avec la λ -abstraction λz pour atteindre la forme normale. La version de gauche a en revanche une étape de retard et devra encore réduire la deuxième occurrence du radical $(\lambda w.w)b$.



Ces mécanismes informels sont complétés chez Wadsworth [Wad71] de critères de correction (d'*admissibilité*) pour les graphes ainsi que d'une notion d'*expression libre* (*abstractable expression*). La prochaine section détaille cette dernière notion.

2.2 Structure du λ -calcul faible

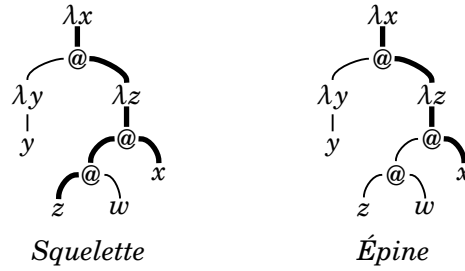
La présente section introduit de nouvelles définitions pour le λ -calcul et établit quelques propriétés élémentaires de la réduction faible qui nous seront utiles à ce chapitre et au suivant pour construire des variantes étiquetées du λ -calcul.

Définition 2.1 (Préfixe). Une **préfixe** n -aire d'un λ -terme t est un λ -contexte n -aire c tel qu'il existe des termes t_1, \dots, t_n satisfaisant $t = c[t_1, \dots, t_n]$. Une **fonction préfixe** est une fonction prenant une λ -abstraction $\lambda x.t$ en entrée et renvoyant un préfixe de $\lambda x.t$.

Deux préfixes appelés *squelette* (définition 2.5) et *épine* (définition 2.4) seront particulièrement utiles aux constructions à venir. Le *squelette* d'une λ -abstraction $\lambda x.t$ est caractérisé négativement par l'ablation des expressions libres de $\lambda x.t$ (la définition des expressions libres, nommées en version originale *abstractable* par Wadsworth est rappelée en définition 0.42). L'*épine* d'une λ -abstraction $\lambda x.t$ est elle caractérisée positivement comme le préfixe constitué des chemins menant aux occurrences liées de x .

Exemple 2.2.

Soit le terme $t = \lambda x.(\lambda y.y)(\lambda z.zwx)$. Le *squelette* et l'*épine* de t sont respectivement les contextes d'arité 2 $\lambda x.\square(\lambda z.z\square x)$ et $\lambda x.\square(\lambda z.\square x)$. Ils sont représentés en gras dans les figures suivantes.



Remarque 2.3 (Squelettes et épines). Comme le font Shivers et Wand [SW10], nous pouvons caractériser positivement le *squelette* par aggrégation d'*épines* : pour obtenir le *squelette* de $\lambda x.t$, nous pouvons commencer avec l'*épine* de $\lambda x.t$ et ajouter récursivement au préfixe obtenu les *épines* de toutes les abstractions contenues dans le préfixe construit jusque là. C'est ce qu'il advient à l'exemple 2.2 avec l'*épine* $\lambda z.z\square\square$ de la λ -abstraction $\lambda z.zwx$.

Cette remarque 2.3 fonde la description des *squelettes* données par la définition 2.5 ci-dessous.

Définition 2.4 (*Épine*). Pour toute variable x , la x -*épine* d'un λ -terme t , notée $\langle t \rangle^x$, est le préfixe de t défini ainsi :

$$\begin{aligned} \langle t \rangle^x &= \square && x \notin \text{fv}(t) \\ \text{Sinon :} \\ \langle x \rangle^x &= x \\ \langle \lambda y.t \rangle^x &= \lambda y.\langle t \rangle^x && y \neq x \\ \langle t_1 t_2 \rangle^x &= \langle t_1 \rangle^x \langle t_2 \rangle^x \end{aligned}$$

L'*épine* d'une λ -abstraction $\lambda x.t$ est son préfixe $\lambda x.\langle t \rangle^x$.

Définition 2.5 (*Squelette*). Pour tout ensemble de variables θ , le θ -*squelette* d'un λ -terme t , noté $\langle\langle t \rangle\rangle^\theta$, est le préfixe de t défini ainsi :

$$\begin{aligned} \langle\langle t \rangle\rangle^\theta &= \square && \theta \cap \text{fv}(t) = \emptyset \\ \text{Sinon :} \\ \langle\langle x \rangle\rangle^\theta &= x && x \in \theta \\ \langle\langle \lambda x.t \rangle\rangle^\theta &= \lambda x.\langle\langle t \rangle\rangle^{\theta \cup \{x\}} \\ \langle\langle t_1 t_2 \rangle\rangle^\theta &= \langle\langle t_1 \rangle\rangle^\theta \langle\langle t_2 \rangle\rangle^\theta \end{aligned}$$

Le **squelette** d'une λ -abstraction $\lambda x.t$ est son préfixe $\lambda x.\langle\langle t \rangle\rangle^{[x]}$. Nous le noterons $\mathcal{P}_W(\lambda x.t)$ par anticipation du chapitre 3.

Les notions de squelette et d'épine seront des éléments importants pour faire émerger la structure du λ -calcul faible. Et pour cause, les épines et les squelettes permettent de décrire les parties des termes qui sont laissées stables par la réduction faible.

Remarque 2.6 (Positions gelées). *Une position d'un λ -terme t est gelée si et seulement si elle appartient à l'épine d'une λ -abstraction de t . Nous savons donc également qu'une position d'un λ -terme t est gelée si et seulement si elle appartient au squelette d'une λ -abstraction de t .*

Lemme 2.7 (Squelettes et substitutions). *Soient une λ -abstraction $\lambda x.t$ et une substitution $\{y := u\}$ telles que $(\lambda x.t)^{\{y := u\}}$ est définie. Alors $\lambda x.t$ et $(\lambda x.t)^{\{y := u\}}$ ont même squelette.*

Démonstration. Si y n'est pas libre dans $\lambda x.t$ alors c'est immédiat. Supposons donc $y \in \text{fv}(\lambda x.t)$. Les occurrences libres de y sont en particulier des expressions libres de $\lambda x.t$ et n'appartiennent pas au squelette. De même, la définition de $(\lambda x.t)^{\{y := u\}}$ implique que les occurrences substituées de u sont libres dans $(\lambda x.t)^{\{y := u\}}$ et n'influent pas sur le squelette. \square

Lemme 2.8 (Invariance des squelettes). *Soient $\rho : t \rightarrow t'$ un pas de β -réduction faible et s' un squelette dans t' de position p' . Notons p l'ancêtre dans t de p' . Alors $t|_p$ est une λ -abstraction et son squelette s est égal à s' .*

Démonstration. Par cas sur la position $p_\rho = \text{root}(\rho)$ de ρ relativement à p .

- Si les positions p_ρ et p sont disjointes alors c'est immédiat.
- Si p est un préfixe de p_ρ , alors p_ρ est soit dans le squelette, soit dans une expression libre de $t|_p$. Le premier cas est interdit par la réduction faible car il implique que p_ρ soit gelée. Dans le deuxième cas le lemme 2.7 conclut.
- Si p_ρ est un préfixe de p , alors soit s est le squelette principal du radical, soit s est strictement inclus dans ce dernier, soit s est inclus dans une expression libre de $t|_{p_\rho}$. Le premier cas contredit l'hypothèse car la racine du squelette principal n'a pas de descendant. Le deuxième cas est résolu par le lemme 2.7. Dans le troisième cas $t|_p$ est égal à ses descendants éventuels. \square

Comme l'épine d'une λ -abstraction est toujours incluse dans le squelette de ladite λ -abstraction, le lemme précédent vaut aussi pour les épines :

Corollaire 2.9 (Invariance des épines). *Soient $\rho : t \rightarrow t'$ un pas de β -réduction faible et s' une épine dans t' de position p' . Notons p l'ancêtre dans t de p' . Alors $t|_p$ est une λ -abstraction et son épine s est égale à s' .*

Une conséquence du lemme 2.8 est le lemme suivant sur les positions gelées :

Lemme 2.10. *L'ancêtre d'une position gelée est une position gelée.*

Une propriété connue du λ -calcul en réduction faible est l'absence de nécessité d'aucun renommage de variables au cours des réductions, pour peu que le terme d'origine ait des variables liées toutes différentes, et distinctes des variables libres [FMS05]. Cette remarque justifie l'usage qui est fait dans cette thèse de termes syntaxiques, pour lesquels nous ne considérons pas le quotient par α -équivalence.

Il convient cependant dans ce cas de noter que la convention de Barendregt sur les variables, qui est usuellement utilisée conjointement au renommage pour éviter tous les phénomènes désagréables de capture de variables, n'est pas préservée par la β -réduction, cette dernière fût-elle faible. L'exemple 2.11 témoigne de ce fait, et motive l'introduction d'une notion locale de la convention de Barendregt (définition 2.12), qui est elle préservée par la β -réduction faible (lemme 2.14) et sera utile en particulier au chapitre 3.

Exemple 2.11 (Rupture de la convention de Barendregt). _____
Considérons le λ -terme

$$(\lambda x.xx)(\lambda yz.yz)$$

qui est clos et dont toutes les λ -abstractions lient des noms différents. Une première étape de réduction donne le terme

$$(\lambda yz.yz)(\lambda yz.yz)$$

dans lequel des λ -abstractions lient les mêmes noms. Ces λ -abstractions occupant des positions disjointes la convention de Barendregt est toujours respectée. Cependant, un deuxième pas de réduction mène au terme

$$\lambda z.(\lambda yz.yz)z$$

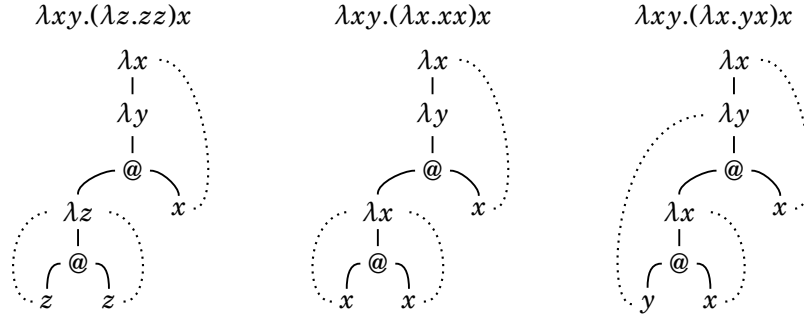
qui ne respecte pas la convention de Barendregt. En effet ce terme contient en particulier le sous-terme $(\lambda yz.yz)z$ dans lequel la variable z apparaît à la fois libre et liée.

Ce sous-terme est un pré- β -radical qui nécessiterait avant réduction le renommage d'au moins une des variables du terme. Mais ce pré- β -radical est avant tout bloqué par la restriction de réduction faible, ce qui rend finalement non-nécessaire ce renommage.

Nous tirerons en revanche avantage au chapitre 3 d'une version locale de la convention de Barendregt, limitée aux squelettes d'un terme.

Définition 2.12 (Convention de Barendregt locale). *Nous dirons qu'un terme t respecte **localement** la convention de Barendregt, noté $\mathbb{B}_l(t)$, si tout squelette de t respecte la convention de Barendregt.*

Exemple 2.13 (Convention de Barendregt locale). _____
Considérons ces trois termes, dans lesquels les lignes pointillées soulignent la liaison des variables.



Le terme de gauche respecte la convention de Barendregt, car toutes ses variables liées sont différentes. Le terme du milieu ne respecte pas la convention de Barendregt à cause de la variable x , mais respecte localement cette convention car ses trois squelettes $\lambda xy.\square x$, $\lambda y.\square$ et $\lambda x.xx$ respectent la convention de Barendregt. Le terme de droite ne respecte pas même localement la convention de Barendregt, à cause du squelette $\lambda xy.(\lambda x.yx)x$ (qui est le terme entier).

Lemme 2.14 (Convention de Barendregt locale). Soit $t \rightarrow t'$ un pas de β -réduction faible. Si $\mathbb{B}_l(t)$ alors $\mathbb{B}_l(t')$.

Démonstration. Immédiat avec le lemme 2.8 d'invariance des squelettes. \square

2.3 Représentation étiquetée des λ -graphes

Nous allons définir ici une variante étiquetée du λ -calcul qui fournira un SLS représentant la technique de réduction de graphes de Wadsworth. Les étiquettes seront un peu plus riches que celles proposées pour l'exemple filé du chapitre 1.

Dans ce précédent exemple nous construisions des étiquettes de la forme $[\omega, p]$ avec ω une étiquette et p une position pour dénoter le symbole à la position p dans le membre droit d'une règle lors de la contraction d'un radical dont l'étiquette racine est ω . Cette approche est liée au fait que les symboles des membres droits de règles sont créés par le pas de réduction correspondant, et qu'il faut donc leur forger des étiquettes fraîches à partir de presque rien.

Dans le λ -calcul en revanche, toute position du réduct d'un pas de réduction descend d'une position de son radical. Nous ne sommes donc pas forcés de créer de nouvelles étiquettes *ex nihilo*, et pouvons réutiliser l'étiquette de l'ancêtre pour construire les étiquettes des descendants ainsi que l'ont fait Blanc, Lévy et Maranget [BLM07]. Dans ce chapitre, nous construirons donc des étiquettes de la forme $[\omega, \alpha]$ avec ω et α deux étiquettes pour dénoter un descendant d'une position étiquetée par α et « affectée » par la contraction d'un radical d'étiquette racine ω . Cette approche est un peu plus simple à définir formellement dans le cas du λ -calcul. Nous utiliserons encore les positions en tant qu'étiquettes initiales. C'est un choix arbitraire mais pratique.

Définition 2.15 (Étiquettes pour le λ -calcul). *L'ensemble d'étiquettes \mathcal{L} est défini par la grammaire suivante :*

$$\alpha ::= p \mid [\alpha, \alpha] \quad p \text{ position}$$

La relation \leq est définie comme le plus petit ordre partiel tel que $\omega \leq [\omega, \alpha]$ pour toutes étiquettes $\alpha, \omega \in \mathcal{L}$.

Définition 2.16 (λ -termes étiquetés). *Soit \mathcal{X} un ensemble dénombrable de variables. Les λ -termes étiquetés sont définis par la grammaire suivante :*

$$t ::= x \mid \lambda^\alpha x.t \mid (tt)^\alpha \quad x \in \mathcal{X}$$

Alternativement, une application étiquetée $(t_1 t_2)^\alpha$ pourra être notée $@^\alpha(t_1, t_2)$. La notion de λ -contexte est étendue à la grammaire étiquetée. Les positions $\text{pos}(c)$ et les variables libres $\text{fv}(c)$ et liées $\text{bv}(c)$ d'un λ -contexte c sont définies comme d'habitude, sans tenir compte des étiquettes. De même pour la substitution et les notions de préfixe, squelette et épine. Comme à la définition 2.5 nous noterons $\mathcal{P}_W(\lambda^\alpha x.t)$ le squelette d'une λ -abstraction étiquetée $\lambda^\alpha x.t$.

À tout λ -terme t peut être associé un λ -terme étiqueté qualifié d'*initial*, obtenu en étiquetant chaque λ -abstraction et chaque application de t par sa position.

Définition 2.17 (Étiquetage initial). *Pour toute position p la fonction $i_p(\cdot)$ transforme un λ -terme en un λ -terme étiqueté, avec les règles suivantes :*

$$\begin{aligned} i_p(x) &= x \\ i_p(\lambda x.t) &= \lambda^p x.i_{p-1}(t) \\ i_p(t_1 t_2) &= @^p(i_{p-1}(t_1), i_{p-2}(t_2)) \end{aligned}$$

Soit t un λ -terme. L'étiquetage initial de t est le λ -terme étiqueté $i_c(t)$, que nous notons encore $i(t)$.

Définition 2.18 (Réétiquetage uniforme). *La composition d'étiquettes est étendue aux λ -termes étiquetés et aux λ -contextes étiquetés : $[\omega, c]$ dénote le λ -contexte étiqueté c dans lequel toute étiquette α a été remplacée par $[\omega, \alpha]$. Formellement :*

$$\begin{aligned} [\omega, \square] &= \square \\ [\omega, x] &= x \\ [\omega, \lambda^\alpha x.c] &= \lambda^{[\omega, \alpha]} x.[\omega, c] \\ [\omega, @^\alpha(c_1, c_2)] &= @^{[\omega, \alpha]}([\omega, c_1], [\omega, c_2]) \end{aligned}$$

Définition 2.19 (Pré- β -radical étiqueté). *Un pré- β -radical étiqueté est un λ -terme étiqueté de la forme $((\lambda^\alpha x.t)u)^\omega$.*

Définition 2.20 (β -réduction faible étiquetée). *Soit $((\lambda^\alpha x.t)u)^\omega$ un pré- β -radical. Notons $t = s[t_1, \dots, t_n]$ avec $\lambda^\alpha x.s = \mathcal{P}_W(\lambda^\alpha x.t)$. Nous avons alors la règle de réduction β_l :*

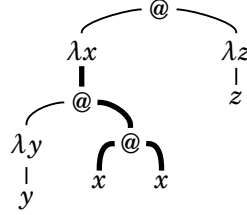
$$((\lambda^\alpha x.s[t_1, \dots, t_n])u)^\omega \rightarrow_{\beta_l} [\omega, s]^{[x:=u]}[t_1, \dots, t_n]$$

La β -réduction faible étiquetée est la réduction faible (combinatoire) générée par le schéma de règle β_l .

Exemple 2.21.

Considérons le pré- β -radical étiqueté

$$@^\alpha(\lambda^\beta x. @^\gamma(\lambda^\delta y. y, @^\zeta(x, x)), \lambda^\eta z. z)$$



Le squelette de la λ -abstraction principale (en gras ci-dessus) est

$$\lambda^\beta x. @^\gamma(\square, @^\zeta(x, x))$$

et le réduit est

$$@^{[\alpha, \gamma]}(\lambda^\delta y. y, @^{[\alpha, \zeta]}(\lambda^\eta z. z, \lambda^\eta z. z))$$

Définition 2.22 (Propriété des squelettes arborescents). *Nous dirons qu'un λ -terme étiqueté t vérifie la propriété des squelettes arborescents, noté $\mathbb{F}(t)$, si pour tout squelette s dans t , les étiquettes de s sont deux à deux distinctes.*

Le lemme 2.8 peut être exprimé comme suit pour la réduction étiquetée faible.

Lemme 2.23 (Invariance des squelettes). *Soit $t \rightarrow t'$ un pas de β -réduction étiquetée faible dont le radical a pour étiquette racine ω , et soit s' un squelette dans t' . Notons s le squelette de t ancêtre de s' . Alors $s' = s$ ou $s' = [\omega, s]$.*

Démonstration. Similaire à la démonstration du lemme 2.8. □

Ce lemme 2.23 a pour corollaire immédiat :

Lemme 2.24 (Squelettes arborescents). *Soit $t \rightarrow t'$ un pas de β -réduction étiquetée faible. Si $\mathbb{F}(t)$ alors $\mathbb{F}(t')$.*

Avec cet invariant en tête nous pouvons représenter la β -réduction étiquetée comme un SLS. Comme mentionné dans la convention 0.37 l'opérateur de λ -abstraction est représenté par un symbole unaire λx pour chaque variable x . Par habitude nous noterons $\lambda^\alpha x$ plutôt que λx^α la version étiquetée du symbole λx .

Instanciation 2.25. *Considérons le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- la signature $\mathcal{S} = \mathcal{X} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{@\}$,
- l'ensemble \mathcal{T} des λ -termes étiquetés respectant la propriété des squelettes arborescents, et

– l'ensemble \mathcal{R} des pas de β -réduction faible étiquetée dont la source respecte la propriété des squelettes arborescents.

Pour tout pas de réduction $\rho \in \mathcal{R}$, notons

$$\rho : c[(\lambda^\alpha x.s[t_1, \dots, t_n]u)^\omega] \rightarrow_{\beta_i} c[[\omega, s]^{x:=u}[t_1, \dots, t_n]]$$

avec $\mathcal{P}_W(\lambda^\alpha x.s[t_1, \dots, t_n]) = \lambda^\alpha x.s$, et définissons :

- $\text{dex}(\rho) = ((\lambda^\alpha x.s[t_1, \dots, t_n]u)^\omega)$
- $\text{duct}(\rho) = [\omega, s]^{x:=u}[t_1, \dots, t_n]$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \text{pos}(s) \setminus \text{fo}_x(s)$

Lemme 2.26. *Le système $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration.

Source & Cible Par lemme 2.24.

Résidu Par lemme 2.10 l'ancêtre d'une position gelée est gelée. Donc tout descendant d'un radical (non gelé) est toujours un radical.

Zone d'effet Soit un pas de réduction

$$\rho : c[(\lambda^\alpha x.s[t_1, \dots, t_n]u)^\omega] \rightarrow_{\beta_i} c[[\omega, s]^{x:=u}[t_1, \dots, t_n]]$$

avec $\mathcal{P}_W(\lambda^\alpha x.s[t_1, \dots, t_n]) = \lambda^\alpha x.s$. Soit $p \in \text{pos}([\omega, s]^{x:=u}[t_1, \dots, t_n]) \setminus \text{effz}(\rho)$. Comme $\text{effz}(\rho) = \text{pos}(s) \setminus \text{fo}_x(s)$ la position p désigne soit un sous-terme de u soit un sous-terme d'un des t_i . \square

Lemme 2.27. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine La racine de tout radical est une application, et toutes les applications sont étiquetées.

Progression Toute étiquette de la zone d'effet d'une réduction ρ est de la forme $[\omega, \alpha]$ avec $\omega = \tau(\rho)$. Et en particulier $\omega < [\omega, \alpha]$.

Héritage Soit un pas de réduction $\rho \in \mathcal{R}$ d'étiquette racine $\tau(\rho) = \omega$. Soit $\beta \in \mathcal{L}$ une étiquette apparaissant dans la zone d'effet. Alors β est de la forme $[\omega, \gamma]$. En particulier toute étiquette $\alpha < \beta$ vérifie $\alpha \leq \omega$, et même $\alpha < \omega$ si $\alpha \neq \omega$.

Partage La zone d'effet d'une réduction est de la forme $[\omega, s]$ où s est inclus dans un squelette de la source. Par propriété des squelettes arborescents, les étiquettes de deux positions différentes de la zone d'effet ne peuvent être égales. \square

Ainsi la réduction parallèle étiquetée du λ -calcul étiqueté faible présenté dans cette section simule un système de réduction de graphes très similaire à celui de Wadsworth. La discussion de la section 2.4 suivante détaille les différences.

2.4 Discussion : appel par nécessité

2.4.1 Stratégies de réduction

La technique de réduction de graphes de Wadsworth [Wad71] a été définie en association avec une stratégie de réduction en appel par nom. La réduction de graphes que nous avons représentée dans ce chapitre reproduit les mêmes mécanismes de partage et de duplication, mais avec deux différences du point de vue des stratégies de réduction :

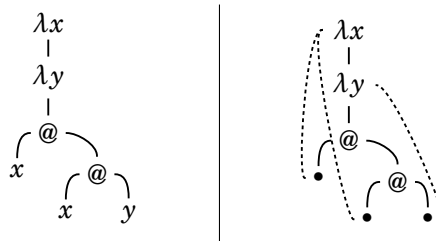
- Nous avons travaillé en réduction faible, et nous sommes donc limités à la réduction jusqu'à une forme normale de l'espace de réduction faible, qui peut être différente de la forme normale de l'espace de réduction non-faible.
- Nous avons autorisé toutes les réductions partagées faibles, ce qui nous permet de couvrir d'autres stratégies de réduction que l'appel par nom.

Cette ouverture à toutes les stratégies de réduction faibles nous permettra au chapitre 10 de raisonner sur les stratégies de réduction optimales. La restriction à la réduction faible en revanche fait que nous n'implémentons pas jusqu'au bout l'algorithme de Wadsworth. Nous leverons cette restriction au chapitre 11.

2.4.2 Représentation des variables liées

Les graphes originellement proposés par Wadsworth et ceux représentés par des termes étiquetés diffèrent dans leur représentation des variables liées.

Avec les termes étiquetés nous avons utilisé des noms pour les variables, comme il est usuel en λ -calcul. Dans ses λ -graphes en revanche, Wadsworth représente une occurrence de variable liée par un *pointeur arrière* conduisant au lieu de cette occurrence. Ces pointeurs sont figurés ci-dessous par les lignes pointillées.



Cette technique permet de se passer des noms de variables et d'éviter toute considération de capture de variables ou de convention de Barendregt. Cette vision des variables peut être recouverte en utilisant une propriété de partage \mathbb{S} moins syntaxique que nous discuterons au chapitre 3.

Bilan du chapitre

Dans ce chapitre nous avons d'abord présenté informellement la technique de Wadsworth pour la réduction efficace du λ -calcul, qui introduit l'appel par

nécessité et le partage pleinement paresseux grâce à la représentation des λ -termes par des graphes. Nous avons ensuite construit un système étiqueté *SLS* spécifiant cette technique d'évaluation (définition 2.25). Ce système étiqueté servira d'inspiration à la fois à l'étiquetage paramétrique du λ -calcul présenté au chapitre 3 et à l'étiquetage des *CRS* défini au chapitre 4.

Pour préparer cette construction nous avons observé quelques propriétés de la β -réduction faible. En particulier nous avons isolé deux briques élémentaires du λ -calcul faible qui seront amenées à resservir par la suite : les notions d'épine et de squelette d'une λ -abstraction (définitions 2.4 et 2.5). En outre, nous avons démontré que les épines et squelettes sont stables par β -réduction faible (lemme 2.8) et en avons déduit la stabilité d'une version locale de la convention de Barendregt 2.12 qui sera utile dans le développement du prochain chapitre.

CHAPITRE 3

Les héritiers de Wadsworth

Sommaire

3.1	Tour d’horizon de la littérature	98
3.2	Un étiquetage paramétrique pour le λ-calcul	101
3.3	Retour à la littérature	107
3.4	La famille pleinement paresseuse	110
3.5	Discussion : partage	112

Les 40 dernières années ont vu naître à intervalles réguliers plusieurs travaux faisant référence à tout ou partie des techniques utilisées par Wadsworth et se réclamant de l’appel par nécessité ou de la pleine paresse.

Ces travaux présentent des différences de forme conséquentes l’un par rapport à l’autre, et en particulier par rapport à la technique d’origine. Autre fait, qui est peut-être conséquence du premier, les comparaisons rigoureuses entre ces systèmes manquent pour beaucoup. Il n’a donc pas été établi formellement que ces différents travaux, quoique de formes différentes, décrivent bien comme souhaité le même fond.

Le présent chapitre comble en grande partie cette lacune avec une comparaison formelle de cinq [Wad71, AF97, Ses97, SW10, BLM07] des sept systèmes majeurs pour l’évaluation pleinement paresseuse dont j’aie connaissance. Un sixième système [PJ87] sera ajouté à cette comparaison au chapitre 8. Le septième système [Tur79] transforme l’espace de réduction d’une façon telle que la comparaison ne pourra être menée ici.

La section 3.1 entame ce chapitre en présentant les différents systèmes qualifiés de pleinement paresseux dans la littérature. La section 3.2 généralise le λ -calcul étiqueté du chapitre 2 de manière à nous permettre de construire en section 3.3 des représentations homogènes de cinq systèmes pleinement paresseux. Nous verrons à cette occasion que ces approches ne correspondent pas toutes au même système de réduction de graphes. La section 3.4 prouvera cependant que les cinq systèmes considérés sont fortement équivalents, dans le sens qu’ils définissent des réductions partagées bisimilaires. La section 3.5 enfin discutera de généralisations non acceptables de l’étiquetage du chapitre 2, ainsi que de variations possibles sur la propriété de partage.

3.1 Tour d’horizon de la littérature

L’évaluation partagée du λ -calcul, et en particulier l’évaluation partagée pleinement paresseuse, apparaît sous différentes formes dans la littérature. Nous commencerons dans cette section par exposer quelques techniques générales utilisées pour exprimer le partage de sous-termes (section 3.1.1) puis verrons leur application au cas du partage pleinement paresseux (section 3.1.2).

3.1.1 Techniques générales pour le partage de sous-termes

Nous avons déjà vu dans les chapitres précédents deux outils utiles à la formalisation de la réduction partagée : les graphes, et leur représentation par des termes étiquetés. Deux autres outils sont encore couramment utilisés.

Extraction des paramètres libres. Dans des graphes tels que les λ -graphes de Wadsworth [Wad71] apparaissent les constructions syntaxiques du λ -calcul et en particulier les lieurs. Ceci nécessite soit de définir le renommage de variables dans les graphes, soit d’y ajouter d’autres structures représentant les liaisons (c’est cette deuxième option qui est formalisée par Wadsworth). Dans tous les cas, le système obtenu est plutôt complexe. La réduction de graphes peut être rendue plus simple en transformant les λ -termes (qui sont des termes d’ordre supérieur) en expressions applicatives (qui sont des termes du premier ordre). Cette transformation est l’objet de la compilation du λ -calcul dans des combinateurs [Tur79] ou supercombineurs [Hug82], qui a donné naissance à la transformation de programmes appelée *λ -lifting* [Joh85, PJ87].

Cette transformation extrait toutes les variables libres d’une fonction et remplace ce qui reste de la fonction par un nouveau symbole appelé supercombinateur. De nouvelles règles de réduction donnant la dynamique de ces nouveaux symboles sont alors ajoutées.

Fermetures et tas mémoire. Alors que les graphes sont un moyen simple et éprouvé de décrire le partage, le système de référence pour la sémantique de l’évaluation paresseuse est la sémantique naturelle de Launchbury [Lau93]. Ce système utilise des constructions `let... in...` pour nommer les arguments d’une fonction avant de les placer dans un tas mémoire. Le partage apparaît ainsi comme une *mémorisation* : quand un évaluateur a besoin de la valeur d’une variable, il évalue l’expression correspondante trouvée dans le tas mémoire, puis met à jour le tas mémoire avec le résultat obtenu.

Inversement à l’approche *grands pas* de Launchbury, Ariola et al. donnent de l’évaluation paresseuse une description à *petits pas* à base de termes dans leur système appelé « λ -calcul en appel par nécessité » [AFM⁺95, AF97, MOW98, CF12]. Ici encore, le partage est exprimé à l’aide de constructions `let... in...` additionnelles utilisées comme des fermetures. Une variante d’esprit assez proche exprime le partage à l’aide de substitutions explicites [Yos94, Ros96].

Ces deux styles *petits pas* et *grands pas* pour la réalisation d'idées similaires peuvent être mieux compris par leur mise en relation étroite au moyen de transformations de programmes bien connues [DMMZ10].

3.1.2 Tour d'horizon de la pleine paresse

Rappelons-nous l'idée directrice du partage « pleinement paresseux » : les parties *constantes* d'une fonction ne sont par définition pas affectées par l'instanciation de ladite fonction, et il n'est donc pas nécessaire de les dupliquer à cette occasion.

Les différentes définitions du partage pleinement paresseux combinent cette idée avec un ou plusieurs des mécanismes décrits en section 3.1.1. Nous obtenons une variété de visions du partage pleinement paresseux, avec différentes interprétations de la même idée directrice et diverses combinaisons des mécanismes de partage parfois difficilement comparables. Ces approches sont énumérées ci-dessous et résumées par la table 3.1.

1971. La première description du partage pleinement paresseux est la technique d'évaluation de *graphes* de Wadsworth [Wad71]. Comme nous l'avons vu au chapitre 2, cette technique ne duplique que partiellement le corps des λ -abstractions en épargnant la copie des expressions libres maximales (voir aussi exemple 3.1(b)).

1979. Turner [Tur79] propose une amélioration de la transformation classique du λ -calcul en logique combinatoire [CF74]. L'idée de base de l'amélioration est d'empêcher la propagation de substitutions dans des sous-termes où les variables substituées n'ont pas d'occurrences libres. Dans un contexte de réduction de graphes ce phénomène est réputé lié à la pleine paresse.

1982. À la suite de Hughes [Hug82], Peyton-Jones obtient un formalisme de graphes simple en utilisant une version pleinement paresseuse du λ -lifting [PJ87]. Ce λ -lifting pleinement paresseux remplace l'extraction des variables libres d'une fonction par l'*extraction* de ses expressions libres maximales.

1997. Ariola et Felleisen [AF97] et Sestoft [Ses97] utilisent l'*extraction* d'expressions libres pour construire des versions pleinement paresseuses respectivement du λ -calcul en appel par nécessité [AFM⁺95] et de la sémantique naturelle de Launchbury [Lau93]. Dans les deux cas cette extraction est matérialisée par des constructions `let...in...` Il est à noter que la première solution [AF97] utilise une notion d'expression libre légèrement différente et un peu plus restrictive que celle donnée en définition 0.42. Elle est donc susceptible dans certains cas de dupliquer plus de nœuds que les autres (voir exemple 3.1(c)).

	Dupl.	Mécanismes			
		Graphes	Extraction	Fermetures	Étiquettes
[Wad71]	=	✓			
[Tur79]	=	✓	✓		
[PJ87]	=	✓	✓		
[AF97]	Plus		✓	✓	
[Ses97]	=		✓	✓	
[SW10]	=	✓			
[BLM07]	Moins	✓			✓

Le symbole = signifie “autant de nœuds dupliqués que l’algorithme de Wadsworth [Wad71]”.

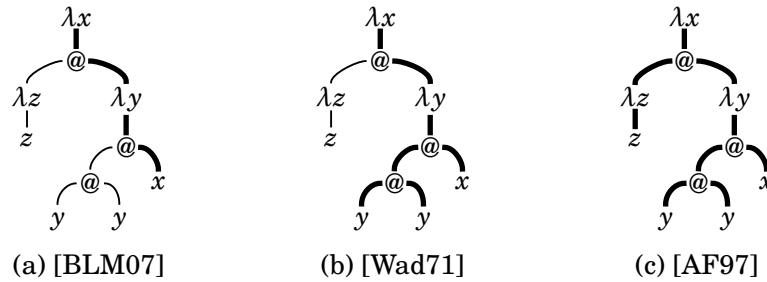
TABLE 3.1: Différentes variations sur la pleine paresse

2005. Shivers et Wand enrichissent la structure des graphes décrits par Wadsworth [Wad71] pour en obtenir une implémentation à la fois simple et efficace [SW10]. Leur algorithme utilise la caractérisation positive des squelettes comme combinaisons d’épines ainsi que décrit dans la remarque 2.3.

2007. Enfin, Blanc, Lévy et Maranget [BLM07] dérivent une implémentation en graphes du partage pleinement paresseux à partir d’étiquettes construites pour l’étude du partage optimal en réduction faible (combinatoire) du λ -calcul. Cette approche est susceptible de dupliquer moins de nœuds d’une λ -abstraction dupliquée (voir exemple 3.1(a)).

Exemple 3.1.

Les lignes grasses indiquent les parties dupliquées de la λ -abstraction suivante dans différents modèles. Voir la section 3.3 pour une définition formelle.



Une contribution de ce chapitre est la définition d’un ensemble de *SLS* sur le λ -calcul dans lesquels seront exprimés les systèmes de réduction de graphes correspondant aux différentes visions de la pleine paresse citées ci-dessus, exceptées celles qui reposent sur le λ -lifting et/ou les combinateurs et supercombinateurs. Autrement dit, cet ensemble de systèmes d’ordre supérieur couvre les approches du partage pleinement paresseux qui travaillent directement à l’ordre supérieur [Wad71, AF97, Ses97, SW10, BLM07].

Exprimer ces différentes approches du partage pleinement paresseux dans un même formalisme de partage étiqueté nous permettra de les comparer formellement pour atteindre les deux conclusions suivantes :

- Les approches décrites ci-dessus correspondent à au moins trois systèmes de réduction de graphes différents. Cela signifie qu’elles n’impliquent pas à strictement parler le même partage. Ainsi, même si ces approches veulent implémenter la même idée directrice, leur équivalence est sujette à caution.
- En revanche, toutes ces approches ont le même espace de réduction. Cela signifie que les différentes implémentations de la pleine paresse effectuent toujours le même nombre de β -réductions. Par conséquent, toute comparaison future entre ces approches peut tranquillement ignorer ce paramètre. Ce point est l’objet du théorème 3.26 de partage équivalent et est le principal résultat de ce chapitre.

Les approches restantes fondées sur une traduction au premier ordre par λ -lifting [Hug82, PJ87] seront traitées au chapitre 8. La traduction au premier ordre de Turner [Tur79] au moyen de combinateurs n’est pas intégrée à cette comparaison, car elle simule des substitutions explicites et introduit des étapes de réduction supplémentaires.

3.2 Un étiquetage paramétrique pour le λ -calcul

Dans cette section nous définissons un ensemble de *SLS* basés sur le λ -calcul généralisant le calcul étiqueté du chapitre 2. Cet ensemble permettra de représenter plusieurs systèmes de réduction de graphes, et en particulier des systèmes pleinement paresseux. D’un point de vue axiomatique, nous passerons d’un système à l’autre en modifiant la zone d’effet (définition 1.9) de la β -réduction étiquetée. D’un point de vue pragmatique, nous agirons sur la fonction préfixe, et \mathcal{P}_W en sera un cas particulier.

Les fonctions préfixes auront des propriétés différentes selon qu’elles attribueront aux λ -abstractions des préfixes plus petits ou plus grands que les squelettes de ces dernières. Nous définissons donc conjointement deux catégories de fonctions préfixes qui donneront naissance à deux catégories de systèmes étiquetés pour la β -réduction.

Définition 3.2 (Fonctions préfixes).

Une **fonction préfixe parcimonieuse** est une fonction préfixe \mathcal{P} telle que :

- Pour toute λ -abstraction $\lambda x.t$ le préfixe $\mathcal{P}(\lambda x.t)$ contient l’épine de $\lambda x.t$ et est contenu dans le squelette de $\lambda x.t$.

Une **fonction préfixe généreuse** est une fonction préfixe \mathcal{P} telle que :

- Pour toute λ -abstraction $\lambda x.t$ le préfixe $\mathcal{P}(\lambda x.t)$ contient le squelette de $\lambda x.t$, et contient en entier tout squelette d’un sous-terme de t dont il contient la racine.

Remarque 3.3. La fonction \mathcal{P}_W du chapitre 2 est l’unique fonction préfixe qui soit à la fois parcimonieuse et généreuse.

Définition 3.4 (Étiquettes et termes). *Nous utilisons dans ce chapitre les mêmes étiquettes et les mêmes termes étiquetés qu'au chapitre 2, définitions 2.15 à 2.18.*

La notion de β -réduction étiquetée du chapitre 2 est directement généralisée en utilisant une fonction préfixe parcimonieuse ou généreuse quelconque.

Définition 3.5 (Bêta-réduction étiquetée). *Soit \mathcal{P} une fonction préfixe parcimonieuse ou une fonction préfixe généreuse. Soit $((\lambda^\alpha x.t)u)^\omega$ un pré- β -radical. Notons $t = s[t_1, \dots, t_n]$ avec $\lambda^\alpha x.s = \mathcal{P}(\lambda^\alpha x.t)$. Alors nous avons la règle β_l de β -réduction étiquetée :*

$$((\lambda^\alpha x.s[t_1, \dots, t_n])u)^\omega \rightarrow_{\beta_l} [\omega, s]^{x:=u}[t_1, \dots, t_n]$$

La relation de β -réduction étiquetée est la réduction faible (combinatoire) générée par le schéma de règle β_l .

Définition 3.6 (Bêta-systèmes). *Un β -système **parcimonieux** (resp. **généreux**) est un λ -calcul étiqueté généré par une fonction préfixe parcimonieuse (resp. généreuse).*

Tout β -système peut être représenté par un SLS. En revanche, la preuve de l'axiome de partage est basée sur des invariants différents sur les termes étiquetés selon que le β -système considéré est parcimonieux ou généreux. La section 3.2.1 traite les β -systèmes parcimonieux et la section 3.2.2 les β -systèmes généreux.

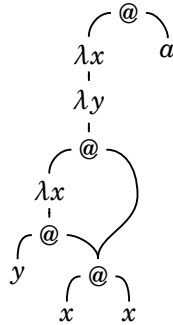
3.2.1 SLS pour les β -systèmes parcimonieux

Les systèmes β -parcimonieux ne modifient que partiellement les étiquettes d'un squelette donné. Par conséquent ils ne préservent pas la propriété des squelettes arborescents (définition 2.22) dans le cas général. En particulier, il n'est pas impossible a priori qu'apparaissent au cours d'une réduction des termes étiquetés comme celui considéré dans l'exemple 3.7 suivant.

Exemple 3.7.

Considérons le terme étiqueté

$$@^\omega(\lambda^\alpha x.\lambda^\beta y.@^\gamma(\lambda^\delta x.@^\zeta(y, @^\eta(x, x)), @^\eta(x, x)), a)$$



qui vérifie la propriété de partage mais pas la propriété des squelettes arborescents. Ce terme est un pré- β -radical. En prenant comme préfixe le squelette de la λ -abstraction principale (c'est-à-dire la totalité de la λ -abstraction) le réduit est

$$\lambda^{[\omega, \beta]} y. @^{[\omega, \gamma]} (\lambda^{[\omega, \delta]} x. @^{[\omega, \zeta]} (y, @^{[\omega, \eta]} (x, x)), @^{[\omega, \eta]} (a, a))$$

L'apparition dans le réduit des deux sous-termes différents $@^{[\omega, \eta]} (x, x)$ et $@^{[\omega, \eta]} (a, a)$ portant la même étiquette $[\omega, \eta]$ met en défaut la propriété de partage.

Dans cet exemple le partage est cassé par la mécanique suivante :

- La fonction contient deux occurrences partagées du terme $@^\eta(x, x)$.
- Bien que les deux sous-termes soient syntaxiquement égaux, les occurrences de la variable x qu'ils contiennent sont liées par des λ -abstractions différentes $\lambda^\alpha x$ et $\lambda^\delta x$, dont l'une est la λ -abstraction principale du radical et pas l'autre.
- Les deux sous-termes étant dans le squelette de la fonction, leur étiquette η est modifiée de la même façon en $[\omega, \eta]$ alors même que l'un est affecté par la substitution $\{x := a\}$ et pas l'autre.

Une manière directe et usuelle de griffer une telle mécanique consisterait à imposer à nos λ -termes de respecter la convention de Barendregt. Las, nous ne pouvons faire ceci sans α -conversion. Nous nous servons donc ici d'un autre invariant de la réduction faible : le respect local de la convention de Barendregt (définition 2.12, lemme 2.14). Cet invariant restreint sera suffisant du fait que l'axiome de partage ne concerne qu'un préfixe inclus dans le squelette de la λ -abstraction principale du β -radical concerné.

Il n'est pas dit en revanche que ce respect local de la convention de Barendregt soit l'invariant le plus général permettant la vérification de l'axiome de partage. Notamment, si je ne suis pas en mesure d'exclure le terme de l'exemple 3.7 je ne sais pas non plus le construire à partir d'un terme initial.

Hormis l'utilisation de ce nouvel invariant, la définition d'un *SLS* et la vérification des axiomes seront ici identiques à ce qui a été fait au chapitre 2. Les points qui diffèrent sont donc exclusivement :

- La définition de l'ensemble \mathcal{T} des termes bien formés.
- La vérification de l'axiome *Source & Cible*.
- La vérification de l'axiome *Partage*.

Instanciation 3.8. *Considérons un β -système généré par une fonction préfixe parcimonieuse \mathcal{P} . Soit le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- *la signature $\mathcal{S} = \mathcal{X} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{@\}$,*
- *l'ensemble \mathcal{T} des λ -termes étiquetés respectant localement la convention de Barendregt, et*
- *l'ensemble \mathcal{R} des pas de β -réduction faible étiquetée dont la source respecte localement la convention de Barendregt.*

Pour toute réduction $\rho \in \mathcal{R}$, notons

$$\rho : c[(\lambda^\alpha x. s[t_1, \dots, t_n]u)^\omega] \rightarrow_{\beta_i} c[[\omega, s]^{[x:=u]}[t_1, \dots, t_n]]$$

avec $\mathcal{P}(\lambda^\alpha x. s[t_1, \dots, t_n]) = \lambda^\alpha x. s$, et définissons :

- $\text{dex}(\rho) = ((\lambda^\alpha x.s[t_1, \dots, t_n])u)^\omega$
- $\text{duct}(\rho) = [\omega, s]^{x:=u}[t_1, \dots, t_n]$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \text{pos}(s) \setminus \text{fo}_x(s)$

Lemme 3.9. *Le système $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration. Les vérifications des axiomes *Résidus* et *Zone d'effet* sont inchangées (voir lemme 2.26).

Source & Cible Par lemme 2.14 de respect local de la convention de Barendregt. \square

Lemme 3.10. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration. Les vérifications des axiomes *Étiquette racine*, *Progression*, et *Héritage* sont inchangées (voir lemme 2.27).

Partage Soit une réduction $\rho : c[(\lambda^\alpha x.s[t_1, \dots, t_n])u]^\omega \rightarrow_{\beta_i} c[[\omega, s]^{x:=u}[t_1, \dots, t_n]] \in \mathcal{R}$, avec $\mathcal{P}(\lambda^\alpha x.s[t_1, \dots, t_n]) = \lambda^\alpha x.s$. Notons $t = s[t_1, \dots, t_n]$, et $t' = [\omega, s]^{x:=u}[t_1, \dots, t_n]$. Supposons $\mathbb{S}(((\lambda^\alpha x.t)u)^\omega)$.

Soient deux positions $p, q \in \text{effz}(\rho)$. En particulier $p, q \in \text{pos}([\omega, s]) = \text{pos}(s)$ et $\tau_p([\omega, s]) = [\omega, \alpha_p]$, $\tau_p(s) = \alpha_p$, $\tau_q([\omega, s]) = [\omega, \alpha_q]$, et $\tau_q(s) = \alpha_q$. Supposons $\tau_p([\omega, s]) = \tau_q([\omega, s])$. En particulier $\alpha_p = \alpha_q$, et $\tau_p(s) = \tau_q(s)$. Par hypothèse $\mathbb{S}(((\lambda^\alpha x.t)u)^\omega)$, donc $s|_p = s|_q$ et $[\omega, s]|_p = [\omega, s]|_q$. Par définition d'un β -système parcimonieux, le préfixe $\lambda^\alpha x.s$ est inclus dans le squelette de $\lambda^\alpha x.t$. Par convention de Barendregt locale appliquée à ce dernier, les occurrences de x dans s sont soit toutes libres dans s , c'est-à-dire toutes liées par la racine $\lambda^\alpha x$, soit toutes liées par des λ -abstractions de s . Donc $[\omega, s]|_p$ et $[\omega, s]|_q$ sont affectées de la même façon par la substitution de x dans $[\omega, s]$, et finalement $t'|_p = [\omega, s]^{x:=u}[t_1, \dots, t_n]|_p = [\omega, s]^{x:=u}[t_1, \dots, t_n]|_q = t'|_q$. \square

Ainsi, tout β -système parcimonieux décrit bien un système de réduction de graphes.

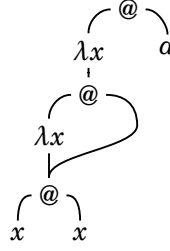
3.2.2 SLS pour les β -systèmes généreux

Pour vérifier l'axiome de partage, au chapitre 2 pour \mathcal{P}_W et à la section 3.2.1 pour les β -systèmes parcimonieux en général, nous avons profité du fait que la zone d'effet était toujours incluse dans la descendance d'un squelette pour lui appliquer des invariants des squelettes (d'abord la propriété des squelettes arborescents puis le respect local de la convention de Barendregt). Ici les zones d'effet sont potentiellement trop grandes pour que ces invariants, quoique toujours vérifiés, soient suffisants.

Exemple 3.11.

Considérons le terme étiqueté

$$@^\omega(\lambda^\alpha x.@^\beta(\lambda^\gamma x.@^\delta(x, x), @^\delta(x, x)), a)$$



qui contient deux squelettes $\lambda^\alpha x. @^\beta(\square, @^\delta(x, x))$ et $\lambda^\gamma x. @^\delta(x, x)$. Le terme vérifie la propriété de partage et la propriété des squelettes arborescents et respecte localement la convention de Barendregt. Il est en outre un pré- β -radical. En prenant comme préfixe toute la λ -abstraction principale, ce qui est bien généreux, le réduit est

$$@^{[\omega, \beta]}(\lambda^{[\omega, \gamma]} x. @^{[\omega, \delta]}(x, x), @^{[\omega, \delta]}(a, a))$$

L'apparition des deux sous-termes $@^{[\omega, \delta]}(x, x)$ et $@^{[\omega, \delta]}(a, a)$ dans le réduit met en défaut la propriété de partage.

La mécanique de cet exemple ressemble fort à celle de l'exemple 3.7. Un point crucial en particulier est l'apparition de deux occurrences partagées d'un sous-terme $@^\delta(x, x)$, qui sont telles que les occurrences de x ne sont pas de part et d'autre liées par la même λ -abstraction.

Un invariant plus fort va nous permettre d'assurer la préservation du partage malgré cette éventualité. L'objectif de cet invariant est d'assurer que dès que deux occurrences d'une variable x sont partagées, alors leur statut de liaison est « cohérent » : soit les deux occurrences sont libres, soit les deux occurrences sont liées, et dans ce dernier cas les deux λ -abstractions concernées portent la même étiquette. Dans un cadre où la propriété de partage est vérifiée cette propriété peut s'exprimer d'une autre façon : dès que deux squelettes partagent un sous-terme, alors ils sont intégralement partagés. (propriété des squelettes accordés, définition 3.13 et lemme 3.14).

Remarquons que cette propriété n'est pas vérifiée par le terme de l'exemple 3.11 car les deux squelettes $\lambda^\alpha x. @^\beta(\square, @^\delta(x, x))$ et $\lambda^\gamma x. @^\delta(x, x)$, bien que partageant un sous-terme $@^\delta(x, x)$, sont bien distincts. En particulier, les occurrences de x sont liées dans l'un par la λ -abstraction $\lambda^\alpha x$ et dans l'autre par la λ -abstraction $\lambda^\gamma x$.

D'abord, remarquons que le lemme 2.23 d'invariance des squelettes est encore valide pour tout β -système généreux.

Lemme 3.12 (Invariance des squelettes). *Soit $t \rightarrow t'$ un pas de β -réduction étiquetée faible, dans un β -système généreux, dont le radical a pour étiquette racine ω , et soit s' un squelette dans t' . Notons s le squelette de t ancêtre de s' . Alors $s' = s$ ou $s' = [\omega, s]$.*

L'invariant suivant nous permettra d'assurer que des sous-termes portant la même étiquette ont des variables liées de la même façon.

Définition 3.13 (Invariant des squelettes accordés). *Un terme étiqueté t a la propriété des **squelettes accordés**, noté $\mathbb{T}(t)$, si deux conditions sont satisfaites :*

- *tout squelette s de t (dont nous rappelons qu'il est un contexte) est de la forme $s = [\omega_n, \dots, [\omega_1, s^*]]$ où toutes les étiquettes de s^* sont initiales, et*
- *pour tous deux squelettes $s_1 = [\omega_n, \dots, [\omega_1, s_1^*]]$ et $s_2 = [\omega'_n, \dots, [\omega'_1, s_2^*]]$ dans t , si s_1^* et s_2^* ont une étiquette en commun alors l'un est un sous-squelette de l'autre.*

Lemme 3.14 (Générosité et accord des squelettes). *Soit $t \rightarrow t'$ un pas de β -réduction étiquetée faible dans un β -système généreux. Si $\mathbb{T}(t)$ alors $\mathbb{T}(t')$.*

Démonstration. Supposons $\mathbb{T}(t)$, et notons ω l'étiquette racine de la réduction.

- Soient s' un squelette dans t' et s son ancêtre dans t . Par hypothèse $s = [\omega_n, \dots, [\omega_1, s^*]]$ où les étiquettes de s^* sont toutes initiales. Nous pouvons donc conclure avec le lemme 3.12.
- Par lemme 3.12, pour tout squelette s' de t' son ancêtre s dans t vérifie $s = s'$ ou $[\omega, s] = s'$. Nous concluons donc avec le deuxième point de $\mathbb{T}(t)$. \square

Instanciation 3.15. *Considérons un β -système généré par une fonction préfixe généreuse \mathcal{P} . Soit le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$, obtenu comme le précédent mais en restreignant les termes :*

- \mathcal{S} est encore l'ensemble $\mathcal{X} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{\@\}$,
- \mathcal{T} est l'ensemble des λ -termes étiquetés respectant localement la convention de Barendregt et vérifiant la propriété des squelettes accordés, et
- \mathcal{R} est l'ensemble des pas de β -réduction faible étiquetée dont la source respecte localement la convention de Barendregt et vérifie la propriété des squelettes accordés.

Les fonctions dex, duct, ctx, et effz sont définies comme précédemment (instanciation 3.8).

Lemme 3.16. *Le système $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration. Les vérifications des axiomes *Résidus* et *Zone d'effet* sont inchangées (voir lemme 2.26).

Source & Cible Par lemme 2.14 de respect local de la convention de Barendregt (qui est toujours vrai avec les étiquettes) et lemme 3.14 des squelettes accordés. \square

Lemme 3.17. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration. Les vérifications des axiomes *Étiquette racine*, *Progression*, et *Héritage* sont inchangées (voir lemme 2.27).

Partage Soit un pas de réduction $\rho : c[(\lambda^\alpha x.s[t_1, \dots, t_n]u)^\omega] \rightarrow_{\beta_l} c[[\omega, s]^{[x:=u]}[t_1, \dots, t_n]] \in \mathcal{R}$, avec $\mathcal{P}(\lambda^\alpha x.s[t_1, \dots, t_n]) = \lambda^\alpha x.s$. Notons $t = s[t_1, \dots, t_n]$ et $t' = [\omega, s]^{[x:=u]}[t_1, \dots, t_n]$, et supposons $\mathbb{S}((\lambda^\alpha x.t)u)^\omega$.

Soient $p, q \in \text{effz}(\rho)$. En particulier $p, q \in \text{pos}([\omega, s]) = \text{pos}(s)$, et $\tau_p([\omega, s]) = [\omega, \alpha_p]$, $\tau_p(s) = \alpha_p$, $\tau_q([\omega, s]) = [\omega, \alpha_q]$, et $\tau_q(s) = \alpha_q$. Supposons $\tau_p([\omega, s]) = \tau_q([\omega, s])$. En particulier $\alpha_p = \alpha_q$. Par hypothèse on a donc $s|_p = s|_q$, et $[\omega, s]|_p = [\omega, s]|_q$. Pour en conclure que $[\omega, s]^{[x:=u]}[t_1, \dots, t_n]|_p = [\omega, s]^{[x:=u]}[t_1, \dots, t_n]|_q$, il suffit maintenant de prouver que $[\omega, s]|_p$ et $[\omega, s]|_q$ sont affectés de la même façon par la substitution de x dans $[\omega, s]$. Si $s|_p$ ne contient pas d'occurrence libre de x c'est immédiat. Supposons donc que $s|_p$ contiennent une occurrence libre de x . Comme x est lié dans $\lambda^\alpha x.s$ il existe une position $p_x \in \text{pos}(s)$ (resp. $q_x \in \text{pos}(s)$) telle que $s(p_x)$ (resp. $s(q_x)$) est une λ -abstraction liant x dans $s|_p$ (resp. $s|_q$). Comme $\tau_p(s) = \tau_q(s)$, par propriété des squelettes accordés nous avons aussi $\tau_{p_x}(s) = \tau_{q_x}(s)$. Par conséquent $s|_{p_x} = s|_{q_x}$ et les positions p_x et q_x sont soit égales soit disjointes. En particulier, soit $p_x = q_x = 11$ (la position de la λ -abstraction principale du radical) et $[\omega, s]|_p$ et $[\omega, s]|_q$ sont tous deux affectés par la substitution de x , soit p_x et q_x sont toutes deux différentes de 11, et ni $[\omega, s]|_p$ ni $[\omega, s]|_q$ n'est affecté par la substitution de x . \square

Ainsi les β -systèmes généreux, de même que les β -systèmes parcimonieux, définissent des systèmes de réduction de graphes.

Remarque 3.18. *Dans la vérification de l'axiome Partage, la preuve que $[\omega, s]|_p$ et $[\omega, s]|_q$ sont affectés de la même façon par la substitution de x dans $[\omega, s]$ n'est nécessaire que parce que nous autorisons ici des termes ne respectant pas la convention de Barendregt sur les variables, ce que nous faisons pour insister sur l'absence de nécessité d' α -conversion dans le λ -calcul faible. En abordant la réduction forte au chapitre 11 nous rétablirons cette convention et simplifierons la vérification de l'axiomatique (instanciation 11.20 et lemme 11.21).*

Remarque 3.19. *Le système \mathcal{P}_W étant à la fois parcimonieux et généreux, chaque nouvel invariant introduit et chaque nouvelle propriété démontrée dans ce chapitre s'applique en particulier à \mathcal{P}_W et renforce notre arsenal pour la manipulation de ce système central.*

3.3 Retour à la littérature

Rappelons le slogan du partage étiqueté : « les étiquettes symbolisent des emplacements mémoire ». Ainsi, le sort des étiquettes lors de la réduction étiquetée décrit directement ce qu'il advient des nœuds du graphe correspondant :

- Une nouvelle étiquette correspond à un nouveau nœud.
- Une étiquette préservée est un nœud qui n'est pas affecté par la réduction.

La clé du codage d'un système de réduction de graphes ou d'un système à base de fermetures dans un *SLS* consiste à modifier exactement les étiquettes de ce qui doit être copié. Comme les étiquettes modifiées dans la cible $\text{tgt}(\rho)$ d'un pas de réduction ρ d'un *SLS* sont exactement les étiquettes contenues dans sa zone d'effet $\text{effz}(\rho)$, ceci revient à définir une zone d'effet couvrant exactement les nœuds à dupliquer.

Les deux approches de Wadsworth [Wad71] et Shivers et Wand [SW10] réalisent un partage pleinement paresseux avec deux implémentations en graphes dans lesquelles la partie dupliquée d'une λ -abstraction est son squelette. Elles utilisent pour la première la définition basée sur les expressions libres maximales et pour la deuxième la caractérisation par épine itérée. Ces deux systèmes de transformation de graphes sont représentés par la fonction préfixe à la fois parcimonieuse et généreuse \mathcal{P}_W que nous avons déjà vue, dans laquelle pour toute λ -abstraction $\lambda x.t$ le préfixe $\mathcal{P}_W(\lambda x.t)$ est le squelette de $\lambda x.t$.

Dans sa révision [Ses97] de la sémantique naturelle de la paresse de Launchbury [Lau93], Sestoft introduit l'idée directrice de la pleine paresse avec une transformation de programmes utilisant des constructions `let...in...` pour extraire les expressions libres maximales : si $\lambda x.s$ est le squelette d'une λ -abstraction $\lambda x.t$ et $\lambda x.t = \lambda x.s[t_1, \dots, t_n]$, alors $\lambda x.t$ est remplacé par `let $x_1 = t_1, \dots, x_n = t_n$ in $\lambda x.s[x_1, \dots, x_n]$` où x_1, \dots, x_n sont des variables fraîches. Une fois faite cette extraction des expressions libres maximales t_1, \dots, t_n de la λ -abstraction $\lambda x.t$, une duplication de la λ -abstraction duplique le sous-terme $\lambda x.s[x_1, \dots, x_n]$ mais ne duplique pas les paramètres t_1, \dots, t_n . Une fois encore, ceci est représenté par la fonction préfixe \mathcal{P}_W .

Dans leur λ -calcul en appel par nécessité [AF97], Ariola et Felleisen permettent la substitution, et donc la duplication, de valeurs. Leur extension dite pleinement paresseuse consiste en la restriction de ces duplications permises à un ensemble de valeurs que nous appellerons « valeurs pleinement paresseuses », c'est-à-dire de valeurs ne contenant aucune expression libre non triviale. Nous ne sommes pas loin ici de retomber sur le cas précédent, mais une différence apparaît car Ariola et Felleisen utilisent une version non-standard de la définition des expressions libres. En effet, ils utilisent le critère de référence [PJ87] rappelé à la définition 0.42, mais excluent de la définition les variables et les λ -abstractions. Le comportement de leur calcul peut être reproduit par la fonction préfixe généreuse \mathcal{P}_{AF} telle que pour toute λ -abstraction $\lambda x.t$, si $\lambda x.s$ est le squelette de $\lambda x.t$ et $\lambda x.t = \lambda x.s[t_1, \dots, t_n]$, alors $\mathcal{P}_{AF}(\lambda x.t) = \lambda x.s[\mathcal{P}_{AF}^\square(t_1), \dots, \mathcal{P}_{AF}^\square(t_n)]$, où \mathcal{P}_{AF}^\square est une fonction auxiliaire définie par :

$$\begin{aligned} \mathcal{P}_{AF}^\square(x) &= \square \\ \mathcal{P}_{AF}^\square(\lambda x.t) &= \mathcal{P}_{AF}(\lambda x.t) \\ \mathcal{P}_{AF}^\square(t_1 t_2) &= \square \end{aligned}$$

Les travaux de Blanc, Lévy et Maranget [BLM07] utilisent déjà un système apparenté à un *SLS*. Leur β -réduction étiquetée modifie uniquement les étiquettes de l'épine de la λ -abstraction principale du radical. Ainsi leur calcul correspond à une fonction préfixe parcimonieuse \mathcal{P}_{BLM} dans laquelle pour toute λ -abstraction $\lambda x.t$ le préfixe $\mathcal{P}_{BLM}(\lambda x.t)$ est l'épine de $\lambda x.t$.

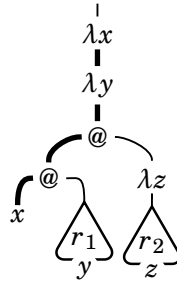
Finalement, les approches d'ordre supérieur du partage pleinement paresseux [Wad71, AF97, Ses97, SW10, BLM07] correspondent à trois β -systèmes définis par trois fonctions préfixes \mathcal{P}_W , \mathcal{P}_{BLM} et \mathcal{P}_{AF} . Comme l'illustre l'exemple 3.1 ces trois systèmes génèrent trois systèmes de réduction de graphes différents. Ce-

pendant, nous allons prouver dans la prochaine section que ces trois approches spécifient la même notion de réduction parallèle étiquetée.

Avant cela, illustrons par un exemple comment les fonctions préfixes \mathcal{P}_{BLM} et \mathcal{P}_{AF} peuvent, malgré leurs différences, générer des réductions parallèles équivalentes.

Exemple 3.20.

Soit $t = \lambda x.\lambda y.xr_1(\lambda z.r_2)$ un terme dans lequel r_1 et r_2 sont deux radicaux dans lesquels la variable x n'apparaît pas libre. Ainsi une duplication de l'épine $\mathcal{P}_{BLM}(t)$ de t , marquée en gras ci-dessous, ne duplique ni r_1 ni r_2 .



Si de plus r_1 (resp. r_2) contient une occurrence libre de la variable y (resp. z), alors les deux radicaux sont au moins partiellement contenus dans $\mathcal{P}_{AF}(t)$, et sont donc dupliqués dans ce système. Mais dans ce cas r_1 et r_2 sont et restent gelés dans chacun des deux systèmes, et leurs étiquettes seront amenées à changer au moins une fois avant le dégel. Ainsi les duplications additionnelles ne sont pas nécessairement dangereuses vis-à-vis de la réduction faible.

Remarque 3.21. Observons enfin que tous les β -systèmes ne sont pas équivalents à l'algorithme de Wadsworth : la fonction préfixe générique \mathcal{P}_N définie par

$$\mathcal{P}_N(\lambda^\alpha x.t) = \lambda^\alpha x.t$$

par exemple implique la duplication intégrale du corps de la λ -abstraction principale de tout β -radical. Le β -système généré par \mathcal{P}_N permet de représenter l'évaluation paresseuse telle que définie en particulier par Launchbury [Lau93] et par Ariola et al. dans la version de base de leur λ -calcul en appel par nécessité [AFM⁺95].

La section suivante commencera donc par définir un ensemble de β -systèmes un peu plus restreint, dont l'objectif est de capturer la pleine paresse. La nouveauté de cette caractérisation tient dans le fait qu'elle ne peut être vue comme une généralisation directe d'aucun des avatars connus de la pleine paresse pris isolément : l'ensemble est issu d'une analyse des similarités et des divergences de ces avatars. Nous donnons ainsi naissance à un nouveau système dont les propriétés peuvent être vues comme l'intersection des propriétés particulières de chaque système pleinement paresseux concret. En d'autres termes, notre axiomatisation tente de saisir l'essence de la pleine paresse telle qu'elle existe actuellement, pour servir de base à la généralisation que porte le chapitre 4.

3.4 La famille pleinement paresseuse

Nous considérons dans cette section un ensemble \mathcal{W} de fonctions préfixes parcimonieuses ou généreuses qui possède deux propriétés :

- L'ensemble \mathcal{W} contient au moins les fonctions préfixes \mathcal{P}_W , \mathcal{P}_{BLM} , et \mathcal{P}_{AF} .
- Les réductions parallèles étiquetées des β -systèmes générés par les fonctions préfixes de l'ensemble \mathcal{W} sont deux à deux bisimilaires.

Définition 3.22 (Ensemble \mathcal{W}). *Une fonction préfixe \mathcal{P} est dans \mathcal{W} si et seulement si pour toute λ -abstraction $\lambda x.t$, le préfixe $\mathcal{P}(\lambda x.t)$ contient exclusivement des positions gelées de $\lambda x.t$.*

Remarque 3.23. *Les fonctions préfixes \mathcal{P}_{BLM} et \mathcal{P}_{AF} ont des positions extrémales dans \mathcal{W} :*

- *La fonction préfixe \mathcal{P}_{BLM} est minimale, dans le sens que pour toute fonction préfixe parcimonieuse ou généreuse \mathcal{P} (dans \mathcal{W} ou non) et toute λ -abstraction $\lambda x.t$, le préfixe $\mathcal{P}(\lambda x.t)$ contient par définition l'épine $\mathcal{P}_{BLM}(\lambda x.t)$. Ceci vaut a fortiori pour toute fonction préfixe $\mathcal{P} \in \mathcal{W}$.*
- *La fonction préfixe \mathcal{P}_{AF} est maximale dans \mathcal{W} : pour toute fonction préfixe $\mathcal{P} \in \mathcal{W}$ et toute λ -abstraction $\lambda x.t$ le préfixe $\mathcal{P}_{AF}(\lambda x.t)$ contient le préfixe $\mathcal{P}(\lambda x.t)$.*

Nous allons maintenant comparer les différents systèmes générés par l'ensemble \mathcal{W} et montrer qu'ils induisent des systèmes de réduction de graphes bisimilaires (théorème 3.26 de partage équivalent). Nous invoquerons pour cette comparaison la notion de termes miroirs introduite au chapitre 1 (définition 1.26). Notons que le lemme 1.27 implique que deux termes miroirs ont les mêmes positions gelées. Le théorème 3.26 de partage équivalent repose sur le lemme suivant.

Lemme 3.24 (Positions non gelées). *Soient un pas de réduction $t \rightarrow t'$ et une position p' de t' . Notons p l'ancêtre dans t de p' et ω l'étiquette racine du radical réduit.*

- *Si p n'est pas gelée, alors p' n'est pas gelée et de plus $\tau_{p'}(t') = \tau_p(t)$.*
- *Si p est gelée mais p' ne l'est pas, alors $\tau_{p'}(t') = [\omega, \tau_p(t)]$.*

Démonstration.

- Par définition les préfixes de les codomaines des fonctions préfixes de \mathcal{W} ne contiennent que des positions gelées. Une position non gelée ne peut donc pas changer d'étiquette.
- Si p est gelée mais un de ses descendants p' ne l'est pas alors p est dans l'épine de la λ -abstraction principale du β -radical réduit. Par définition de \mathcal{W} cette épine est toujours incluse dans le préfixe d'un radical, et par définition de la β -réduction étiquetée $\tau_{p'}(t') = [\omega, \tau_p(t)]$. \square

Chaque terme étiqueté définit une relation d'équivalence sur l'ensemble de ses positions.

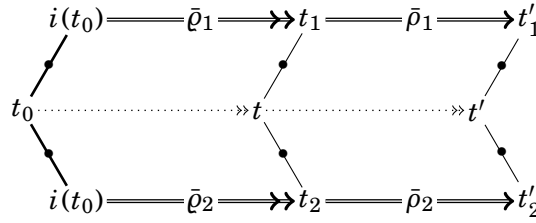
Définition 3.25 (Équivalence par étiquettes). *Soit t un terme étiqueté. Soient $p, q \in \text{pos}(t)$. Les positions p et q sont **équivalentes par étiquettes** si $\tau_p(t) = \tau_q(t)$.*

Théorème 3.26 (Partage équivalent). *Soient \mathcal{P}_1 (resp. \mathcal{P}_2) une fonction préfixe de l'ensemble \mathcal{W} et t_1 (resp. t_2) un λ -terme étiqueté dans le β -système généré par \mathcal{P}_1 (resp. \mathcal{P}_2). Supposons que $t_1 \approx t_2$. Alors t_1 et t_2 induisent la même relation d'équivalence par étiquettes sur les positions non gelées de $t = t_1^\bullet = t_2^\bullet$.*

Démonstration. Par définition de $t_1 \approx t_2$, il existe un terme non étiqueté t_0 et une séquence de réduction miroir parallèle $(\bar{\rho}_1, \bar{\rho}_2)$ de source $(i(t_0), i(t_0))$ et de cible (t_1, t_2) . Raisonnons par récurrence sur le nombre d'étapes de réduction parallèle étiquetée dans la séquence.

Le cas de base, avec une séquence de longueur 0, est immédiat car alors $t_1 = i(t_0) = t_2$.

Supposons donc que la propriété est vérifiée pour les séquences de réduction miroir parallèle de longueur n , et considérons une séquence de réduction miroir parallèle $(\bar{\rho}_1 \bar{\rho}_1, \bar{\rho}_2 \bar{\rho}_2)$ de cible (t'_1, t'_2) , avec $(\bar{\rho}_1, \bar{\rho}_2)$ une séquence de réduction miroir parallèle de longueur n . Notons t_1 et t_2 les cibles respectives de $\bar{\rho}_1$ et $\bar{\rho}_2$.



Par hypothèse de récurrence les termes t_1 et t_2 induisent la même relation d'équivalence par étiquettes sur les positions non gelées de $t = t_1^\bullet = t_2^\bullet$. De plus, u_1 et u_2 vérifient tous deux les propriétés \mathbb{S} , \mathbb{L} , et \mathbb{F} . Notons ω_1 et ω_2 les étiquettes des radicaux réduits en parallèle par $\bar{\rho}_1$ et $\bar{\rho}_2$.

Soient p' et q' deux positions non gelées de t' . Notons p et q leurs ancêtres respectifs dans t . Supposons que $\tau_{p'}(t'_1) = \tau_{q'}(t'_1) = \alpha'$. Nous allons montrer que $\tau_{p'}(t'_2) = \tau_{q'}(t'_2)$. Pour ceci notons $\tau_p(t_1) = \alpha_p$ et $\tau_q(t_1) = \alpha_q$, et raisonnons par cas sur les états de p et q :

- Si p est gelée et pas q , alors par lemme 3.24 nous avons $\alpha_p = \alpha'$ et $[\omega_1, \alpha_q] = \alpha'$. En particulier $\omega_1 < \alpha_p$, ce qui contredit $\mathbb{L}(t_1)$.
- Si ni p ni q n'est gelée, alors par lemme 3.24 nous avons $\alpha_p = \alpha_q = \alpha'$. Par hypothèse de récurrence nous avons de plus $\tau_p(t_2) = \tau_q(t_2)$, et comme les positions ne sont pas gelées $\tau_{p'}(t'_2) = \tau_p(t_2) = \tau_q(t_2) = \tau_{q'}(t'_2)$.
- Si p et q sont toutes deux gelées, alors par lemme 3.24 nous avons $[\omega_1, \alpha_p] = [\omega_1, \alpha_q] = \alpha'$ et en particulier $\alpha_p = \alpha_q$. Donc par hypothèse de récurrence $\tau_p(t_2) = \tau_q(t_2)$. Comme de même p et q sont dégelées par ρ_2 , par lemme 3.24 nous avons $\tau_{p'}(t'_2) = [\omega_2, \tau_p(t_2)] = [\omega_2, \tau_q(t_2)] = \tau_{q'}(t'_2)$.

Symétriquement $\tau_{p'}(t'_2) = \tau_{q'}(t'_2)$ implique $\tau_{p'}(t'_1) = \tau_{q'}(t'_1)$, ce qui conclut la preuve. \square

3.5 Discussion : partage

La propriété de partage, en raison de son énoncé syntaxique, est sensible à la représentation des variables. Les systèmes parcimonieux en particulier jouent sur cette représentation d'une manière qui peut piéger les systèmes généreux, ce que nous verrons à la section 3.5.1. C'est pour cette raison que ces deux catégories ont été traitées séparément à la section 3.2. La section 3.5.2 discute ensuite de l'interaction entre la propriété de partage et le renommage de variables.

3.5.1 Une combinaison pathologique

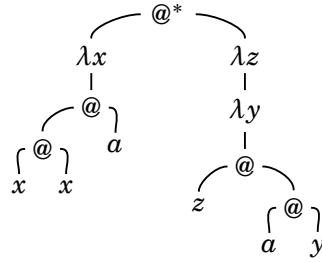
Nous observons dans cette section comment le mélange des spécificités respectives des β -systèmes parcimonieux et des β -systèmes généreux peut être dangereux pour la propriété de partage.

Regardons une réduction dans un β -système imaginaire qui ne serait ni parcimonieux ni généreux. Dans les différents pas de réduction nous utiliserons des préfixes correspondant tantôt à une fonction préfixe parcimonieuse et tantôt à une fonction préfixe généreuse. En partant d'un λ -terme étiqueté initial, nous allons casser la propriété de partage en trois étapes de réduction.

Prenons un λ -terme étiqueté de départ t_0 dont toutes les étiquettes sont initiales (au sens de la définition 1.15) et différentes :

$$t_0 = @^\eta(\lambda^{\epsilon} x . @^{\iota} (@^{\kappa}(x, x), a), \lambda^{\alpha} z . \lambda^{\beta} y . @^{\gamma}(z, @^{\delta}(a, y)))$$

où a est une variable libre. Ce terme t_0 vérifie les invariants \mathbb{S} et \mathbb{I} , et est un radical. Le voici sous forme graphique, avec une étoile pour indiquer la racine du radical qui va être réduit.



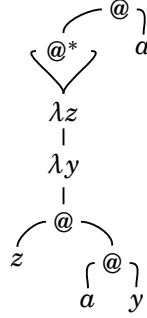
Supposons

$$\mathcal{P}(\lambda^{\epsilon} x . @^{\iota} (@^{\kappa}(x, x), a)) = \lambda^{\epsilon} x . @^{\iota} (@^{\kappa}(x, x), \square)$$

Ce préfixe est ce qu'aurait donné l'une quelconque des trois fonctions \mathcal{P}_W , \mathcal{P}_{BLM} , et \mathcal{P}_{AF} . En posant $\sigma = [\eta, \kappa]$ et $\tau = [\eta, \iota]$ et en appliquant la β -réduction étiquetée usuelle nous obtenons :

$$t_1 = @^{\tau} (@^{\sigma}(\lambda^{\alpha} z . \lambda^{\beta} y . @^{\gamma}(z, @^{\delta}(a, y)), \lambda^{\alpha} z . \lambda^{\beta} y . @^{\gamma}(z, @^{\delta}(a, y))), a)$$

Ce terme t_1 satisfait \mathbb{S} et \mathbb{I} et indique un graphe où les deux occurrences de $\lambda^{\alpha} z . \lambda^{\beta} y . @^{\gamma}(z, @^{\delta}(a, y))$ sont partagées, et où le nœud $@^{\sigma}$ est la racine d'un radical. Graphiquement :



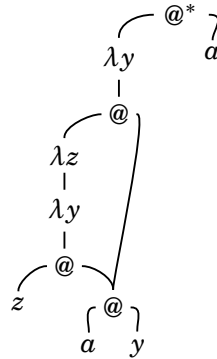
Supposons maintenant

$$\mathcal{P}(\lambda^\alpha z. \lambda^\beta y. @^\gamma(z, @^\delta(a, y))) = \lambda^\alpha z. \lambda^\beta y. @^\gamma(z, \square)$$

Ce préfixe est ce qu'aurait donné la fonction \mathcal{P}_{BLM} , et la réduction correspondante conduit au terme :

$$t_2 = @^\tau(\lambda^{[\sigma, \beta]} y. @^{[\sigma, \gamma]}(\lambda^\alpha z. \lambda^\beta y. @^\gamma(z, @^\delta(a, y)), @^\delta(a, y)), a)$$

Ce terme t_2 satisfait \mathbb{S} et \mathbb{I} et indique un graphe où les deux occurrences de $@^\delta(a, y)$ sont partagées. Ce terme est également un nouveau radical. Graphiquement :



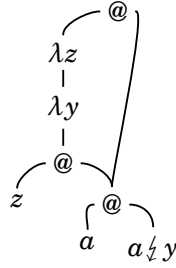
Supposons enfin

$$\begin{aligned} & \mathcal{P}(\lambda^{[\sigma, \beta]} y. @^{[\sigma, \gamma]}(\lambda^\alpha z. \lambda^\beta y. @^\gamma(z, @^\delta(a, y)), @^\delta(a, y))) \\ &= \lambda^{[\sigma, \beta]} y. @^{[\sigma, \gamma]}(\lambda^\alpha z. \lambda^\beta y. @^\gamma(z, @^\delta(\square, y)), @^\delta(\square, y)) \end{aligned}$$

Ce préfixe est ce qu'aurait donné la fonction \mathcal{P}_{AF} . Regardons la cible de l'ultime et fatale réduction associée :

$$t_3 = @^{[\tau, [\sigma, \gamma]]}(\lambda^{[\tau, \alpha]} z. \lambda^{[\tau, \beta]} y. @^{[\tau, \gamma]}(z, @^{[\tau, \delta]}(a, y)), @^{[\tau, \delta]}(a, a))$$

Ce terme étiqueté contient les deux sous-termes $@^{[\tau, \delta]}(a, y)$ et $@^{[\tau, \delta]}(a, a)$ qui invalident la propriété de partage.



Cette chute de la propriété de partage vient de la combinaison des deuxième et troisième pas de β -réduction étiquetée, qui utilisent l'un un préfixe strictement parcimonieux et l'autre un préfixe strictement généreux.

Une particularité du terme t_2 créé après le pas de réduction parcimonieux est la présence de deux occurrences partagées d'une variable y qui sont liées par deux λ -abstractions distinctes $\lambda^{[\sigma, \beta]}y$ et $\lambda^\beta y$. Le préfixe choisi ensuite pour définir la réduction à partir de t_2 recouvre ces deux occurrences de y , dont l'une est liée par la λ -abstraction principale du radical ($\lambda^{[\sigma] \beta}$) et doit donc être substituée, et l'autre est liée par une autre λ -abstraction (λ^β) et ne doit pas être substituée.

Cette liaison de deux occurrences partagées d'une variable par deux lieux différents est tout à fait légitime dans un β -système parcimonieux et n'y pose aucun problème. Cette robustesse des β -systèmes parcimonieux vient du fait que la λ -abstraction λ^β est étrangère au squelette de la λ -abstraction principale ($\lambda^{[\sigma] \beta}$). Elle ne peut donc pas dans un système parcimonieux faire partie du préfixe de la λ -abstraction principale comme c'est le cas dans notre exemple.

3.5.2 Renommage et partage

La réduction pathologique précédente n'aurait pas pu avoir lieu si nous avions imposé à nos systèmes de respecter la convention de Barendregt, comme il est très généralement fait dans les travaux à base de λ -calcul.

Il faut noter cependant que l'observation de cette convention nécessite d'opérer des renommages de variables, comme le montrent le terme t_2 précédent et l'exemple 2.11. Ceci n'est usuellement pas considéré comme un grand mal, car ces renommages sont de toute façon nécessaires à la bonne définition du λ -calcul dans le cas général. Ainsi les définitions de référence appellent *termes* les classes d'équivalence de la relation d' α -conversion dans l'ensemble des *termes syntaxiques*.

Cependant, quand le λ -calcul est restreint à sa réduction faible, le renommage de variables n'est plus nécessaire à la bonne conduite du calcul. Nous avons donc dans ce chapitre accepté comme termes les termes syntaxiques, interdit le renommage, et ainsi abandonné la convention de Barendregt.

Et pour cause, la propriété de partage serait sinon en danger. Prenons par exemple le terme initial suivant :

$$u_0 = @^\alpha(\lambda^\beta x. @^\gamma(x, x), \lambda^\delta y. @^\eta(a, y))$$

Extrayons le seul préfixe admissible pour cette fonction simple :

$$\mathcal{P}(\lambda^\beta x.@^\gamma(x,x)) = \lambda^\beta x.@^\gamma(x,x)$$

Et opérons la réduction, nous obtenons la cible suivante :

$$u_1 = @^{[\alpha,\gamma]}(\lambda^\delta y.@^\eta(a,y), \lambda^\delta y.@^\eta(a,y))$$

Enfin, prenons un λ -terme α -équivalent obtenu en renommant l'une des variables y en z :

$$u_2 = @^{[\alpha,\gamma]}(\lambda^\delta y.@^\eta(a,y), \lambda^\delta z.@^\eta(a,z))$$

Ce terme u_2 contient deux sous-termes $\lambda^\delta y.@^\eta(a,y)$ et $\lambda^\delta z.@^\eta(a,z)$ qui ont la même étiquette mais sont syntaxiquement différents.

La fin de la présente section discute des aménagements possibles pour fluidifier cette interaction entre variables et propriété de partage. La première direction est la libéralisation de la propriété de partage, la deuxième est la modification du système.

Les deux sous-termes $\lambda^\delta y.@^\eta(a,y)$ et $\lambda^\delta z.@^\eta(a,z)$ précédents étant α -équivalents, on pourrait soutenir que le problème est simplement dû à une définition trop syntaxique du partage. Cependant l'exemple contient également les deux sous-termes $@^\eta(a,y)$ et $@^\eta(a,z)$, qui brisent aussi le partage, sans être α -équivalents.

Une version moins syntaxique de la propriété de partage devrait donc aller plus loin, par exemple en remarquant que dans leurs contextes respectifs, les variables y et z présentes dans $@^\eta(a,y)$ et $@^\eta(a,z)$ sont toutes deux liées, et par des λ -abstractions $\lambda^\delta y$ et $\lambda^\delta z$ qui ont même étiquette. Dans un cadre où l'égalité des étiquettes veut être interprétée comme une identité des termes, y et z pourraient être ici considérés comme des variables *suffisamment ressemblantes*. Cette solution fait de plus écho à l'algorithme original de Wadsworth dans lequel les variables liées sont identifiées uniquement par un pointeur arrière vers leur lieu.

Mais là encore le critère doit être discuté. Prenons par exemple ce sous-terme extrait du terme t_2 précédent, qui a été légitimement obtenu dans le système \mathcal{P}_{BLM} que nous savons sûr.

$$t = \lambda^{[\sigma,\beta]} y.@^{[\sigma,\gamma]}(\lambda^\alpha z.\lambda^\beta y.@^\gamma(z,@^\delta(a,y)), @^\delta(a,y))$$

Par renommage d'une des variables y en x , nous restaurons la convention de Barendregt dans ce terme et obtenons :

$$t' = \lambda^{[\sigma,\beta]} x.@^{[\sigma,\gamma]}(\lambda^\alpha z.\lambda^\beta y.@^\gamma(z,@^\delta(a,y)), @^\delta(a,x))$$

Dans ce nouveau terme syntaxique t' cohabitent les deux sous-termes $@^\delta(a,x)$ et $@^\delta(a,y)$. La propriété de partage ne tient pas, et les deux variables x et y , bien que toutes deux liées, le sont par deux abstractions $\lambda^{[\sigma,\beta]}x$ et $\lambda^\beta y$ aux étiquettes et donc aux identités différentes.

Ainsi, pour accepter des systèmes strictement parcimonieux comme \mathcal{P}_{BLM} conjointement avec l' α -équivalence, il faudrait envisager une libéralisation plus importante encore du critère, où seraient par exemple considérées comme *identiques* toutes deux variables liées, quelles que soient les λ -abstractions qui les lient.

Sans changer l'énoncé strict et syntaxique de la propriété de partage, nous pouvons permettre un renommage inoffensif des variables en rendant cette opération visible. L'idée est qu'une opération explicite de renommage va agir sur les étiquettes de manière à préserver la cohérence du terme étiqueté. Nous pourrions par exemple obtenir au lieu du terme t' le terme suivant :

$$t'' = \lambda^{[[\sigma, \beta]]} x. @^{[[\sigma, \beta]] [\sigma, \gamma]} (\lambda^\alpha z. \lambda^\beta y. @^\gamma (z, @^\delta (a, y)), @^{[[\sigma, \beta]] \delta} (a, x))$$

dans lequel le partage entre les sous-termes $@^\delta (a, y)$ et $@^{[[\sigma, \beta]] \delta} (a, x)$ a été annulé par le changement d'étiquettes de l'un des sous-termes pour tenir compte du renommage.

Il serait également envisageable de forcer la réduction à renommer au maximum toutes les variables liées de son réduit afin d'assurer la préservation de la convention de Barendregt. Il y a une précaution à prendre pour cela : pour respecter l'axiome *Zone d'effet*, les renommages doivent avoir lieu à l'intérieur de la zone d'effet. Il faut donc soit être certain que la zone d'effet est suffisamment grande, ce qui est le cas avec le système \mathcal{P}_W , soit étendre la zone d'effet aux parties renommées. Ces deux solutions peuvent être vues comme l'incorporation de la règle de renommage à la règle de β -réduction.

La solution utilisant une règle de renommage explicite sera intégrée à un exemple au chapitre 11.

Bilan du chapitre

Dans ce chapitre nous avons passé en revue les principaux travaux qui, après Wadsworth [Wad71], se sont réclamés de la pleine paresse [Tur79, PJ87, AF97, Ses97, SW10, BLM07]. Afin de plonger de manière uniforme dans le cadre des *SLS* ceux de ces travaux exprimés dans un style proche du λ -calcul [Wad71, AF97, Ses97, SW10, BLM07] nous avons défini une version paramétrée de la β -réduction étiquetée du chapitre 2 (définition 3.5). Pour montrer que les β -systèmes ainsi obtenus forment toujours des *SLS* nous les avons séparés en deux catégories (β -systèmes parcimonieux et généreux, définition 3.2) dont les propriétés sont légèrement différentes et subtilement incompatibles (voir la section 3.5.1).

Parmi tous les β -systèmes nous avons ensuite isolé un ensemble visant à caractériser la notion de pleine paresse (définition 3.22), et avons effectivement montré que d'une part cet ensemble contient les β -systèmes représentant la technique de Wadsworth [Wad71] aussi bien que les autres descriptions de la pleine paresse basées sur le λ -calcul [AF97, Ses97, SW10, BLM07], et d'autre

part que tous les systèmes de cet ensemble produisent les mêmes séquences de réduction parallèle (théorème 3.26 de partage équivalent).

Nous avons enfin remarqué qu'un seul β -système était à la fois parcimonieux et généreux : le système représentant la technique de Wadsworth (remarque 3.3). Par conséquent nous avons un unique β -système présentant à la fois les bonnes propriétés des β -système parcimonieux et des β -systèmes généreux. C'est donc lui qui nous servira naturellement de base au chapitre 4 dans l'élaboration d'un système pleinement paresseux pour la récriture d'ordre supérieur.

CHAPITRE 4

Généralisation à l'ordre supérieur

Sommaire

4.1 Réduction faible dans les CRS	121
4.2 Raffinements	124
4.3 Définition du partage pleinement paresseux	133
4.4 Comparaison avec le λ -calcul étiqueté	136
4.5 Discussion : complexité	139

L'objet de ce chapitre est la généralisation de la notion de partage pleinement paresseux de Wadsworth [Wad71] à la réécriture d'ordre supérieur, ou plus précisément aux *Combinatory Reduction Systems* de Klop [Klo80, KvOvR93].

Pour orienter au mieux cette généralisation faisons le bilan des deux chapitres précédents :

- Le partage pleinement paresseux se décline en λ -calcul sous plusieurs formes [Wad71, AF97, Ses97, SW10, BLM07] qui correspondent à des systèmes de réécriture de graphes différents mais bisimilaires.
- La caractéristique de la pleine paresse est la limitation des duplications à des positions qui sont gelées selon la réduction faible (combinatoire).
- Les β -systèmes pleinement paresseux se divisent en deux catégories, les systèmes parcimonieux et les systèmes généreux, qui vérifient chacune leurs invariants propres.
- L'intersection des systèmes parcimonieux et généreux est un singleton, contenant uniquement le système généré par la fonction préfixe \mathcal{P}_W .

Ainsi le système \mathcal{P}_W , en combinant les bonnes propriétés à la fois des systèmes parcimonieux et des systèmes généreux, est celui qui semble à la fois le plus robuste et le plus aisé à manipuler. Nous donnerons donc raison à Wadsworth et nous baserons sur son système pour la suite. Dans ce cadre nous pourrions également continuer à exploiter les relations entre le partage pleinement paresseux et la réduction faible (combinatoire).

La généralisation portée par ce chapitre est assortie d'une nouveauté métho-

dologique par rapport aux démarches des chapitres 2 et 3, nouveauté que nous détaillons maintenant.

Les étiquetages pleinement paresseux des chapitres 2 et 3 ajoutent au λ -calcul des mécanismes *ad hoc* pour la gestion des étiquettes. Cet ajout répond au fait que les étiquettes sont en supplément de la syntaxe d'origine. Ceci semble inévitable pour le λ -calcul dont la forme est fixée, mais les choses sont différentes dans le cadre plus général et naturellement extensible des *CRS*.

Dans ce chapitre nous étiquetterons des *CRS* comme nous avons étiqueté le λ -calcul aux chapitres 2 et 3, mais nous ferons en sorte que le système étiqueté soit toujours un *CRS*. Ainsi il ne sera pas nécessaire de définir un nouveau cadre étiqueté : les mécanismes d'étiquetage des *CRS* seront *internalisés* dans la théorie générale des *CRS*, sous forme de signatures et de règles ordinaires. Ceci permettra notamment la réutilisation directe de la méta-théorie des *CRS* et des *CRS* orthogonaux pour l'étude de nos *CRS* étiquetés (voir chapitres 7 et 9).

La partie syntaxique de cette internalisation est simple : comme nous l'avons fait pour les *SLS* nous allons remplacer une signature par une signature étiquetée dans laquelle chaque symbole est la combinaison f^α d'un des symboles d'origine f et d'une étiquette α (définition 1.13). Remarquons que cette approche est suggérée par van Oostrom dans sa définition abstraite de l'étiquetage d'un terme du premier ordre [Ter03, Chap. 8], puisque sa définition est basée sur un morphisme entre deux signatures.

La partie dynamique de l'étiquetage est un peu plus délicate et va demander de transformer au préalable les systèmes à étiqueter. Considérons en effet ce qui se passait au chapitre 2 : la version étiquetée de la β -réduction était exprimée par

$$@^\omega(\lambda^\alpha x.s[t_1, \dots, t_n], u) \rightarrow [\omega, s]^{x:=u}[t_1, \dots, t_n]$$

avec la condition $\mathcal{P}_W(\lambda^\alpha x.s[t_1, \dots, t_n]) = \lambda^\alpha x.s$. Ainsi les étiquettes du $\{x\}$ -squelette de $s[t_1, \dots, t_n]$ sont modifiées. Dans les *CRS* la β -réduction est usuellement ([KvOvR93]) exprimée par la règle

$$@(\langle x \rangle Z(x), Z') \rightarrow Z(Z')$$

dans laquelle le $\{x\}$ -squelette du corps de la fonction n'apparaît pas explicitement. Les positions correspondant aux étiquettes modifiées dans le modèle précédent sont en fait ici les positions dynamiques du réduit, au sens de la définition 0.70.

Cependant, dans le cadre de ce chapitre la modification en $[\omega, \alpha]$ de l'étiquette α d'un symbole f correspond à la création d'un nouveau symbole $f^{[\omega, \alpha]}$, et dans un *CRS* les nouveaux symboles ne peuvent apparaître que dans les positions propres du réduit. Il faut donc faire en sorte que la zone d'effet d'un pas de réduction (c'est-à-dire l'endroit où de nouvelles étiquettes apparaissent) soit constituée intégralement de positions propres de la règle *CRS* associée. Ceci peut être obtenu par exemple en transformant la règle β en l'ensemble infini de règles obtenu en instanciant le paramètre s du schéma de règle suivant par un $\{x\}$ -squelette.

$$@^\omega(\langle x \rangle^\alpha .s[Z_1, \dots, Z_n], Z') \rightarrow [\omega, s]^{x:=Z'}[Z_1, \dots, Z_n]$$

Dans une telle règle toutes les positions de s sont des positions propres, et il n'y a pas de positions dynamiques (car toutes les méta-variables sont d'arité 0).

Ce chapitre présente donc d'abord une notion de raffinement d'une règle *CRS*, qui permet de remplacer une règle par un ensemble de règles équivalent et éventuellement infini, et qui permet en particulier de remplacer une règle générale par un ensemble de règles dans lesquelles toutes les méta-variables ont pour arité 0 (ce qui est équivalent au remplacement des positions dynamiques du réduit par des positions propres). Ce raffinement permettra ensuite d'exprimer les *CRS* étiquetés sans sortir du cadre formel des *CRS* et ainsi de bénéficier directement de toute la méta-théorie des *CRS* sur leur version étiquetée.

L'utilisation du raffinement en revanche va faire que l'étiquetage obtenu pour les *CRS* ne correspondra pas exactement à celui défini pour le λ -calcul au chapitre 2. Nous pourrions cependant vérifier que le partage obtenu est le même et que les réductions parallèles étiquetées du λ -calcul du chapitre 2 et du présent chapitre sont bisimilaires (théorème 4.52 de partage équivalent). Ainsi, l'étiquetage des *CRS* proposé ici spécifie bien une généralisation du partage pleinement paresseux de Wadsworth.

Enfin, nous discuterons l'adéquation avec le réel du modèle de l'évaluation avec partage pleinement paresseux. Nous définirons la classe des *CRS plats* (définition 4.64), qui contient en particulier le λ -calcul, et prouverons que la réduction des *CRS plats* peut effectivement être réalisée sur une machine de Turing en un temps polynomial en le nombre d'étapes de réduction dans le modèle avec partage pleinement paresseux (théorème 4.70 de complexité dérivatoire). Autrement dit, la réduction pleinement paresseuse est un modèle raisonnable de l'exécution des programmes fonctionnels.

La section 4.1 commence par transposer aux *CRS* la notion de réduction faible combinatoire et les propriétés de base qui y sont encore vérifiées. La section 4.2 définit ensuite les raffinements d'un ensemble de règles *CRS* et en particulier un raffinement dit canonique qui servira de base à la construction à la section 4.3 de l'étiquetage définissant le partage pleinement paresseux dans les *CRS*. Le bien-fondé de cette vision du partage pleinement paresseux sera ensuite justifié aux sections 4.4 avec la correspondance avec la technique de Wadsworth sur le λ -calcul et 4.5 avec les résultats de complexité.

4.1 Réduction faible dans les *CRS*

Cette section adapte au cadre des *CRS* les définitions et les propriétés de la réduction faible que nous avons déjà vues pour le λ -calcul.

Définition 4.1 (Réduction faible). *La réduction dans un contexte est restreinte au cas suivant : $c[l^\sigma] \rightarrow c[r^\sigma]$ où aucune variable libre du codomaine de la valuation σ sûre pour $l \rightarrow r$ n'est liée par un lieu apparaissant dans c au-dessus du trou.*

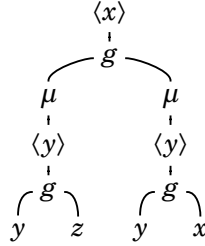
Exemple 4.2.

Considérons un symbole unaire μ . La règle de réécriture

$$\mu(\langle x \rangle Z(x)) \rightarrow Z(\mu(\langle x \rangle Z(x)))$$

décrit un opérateur de point fixe. Soit

$$t = \langle x \rangle g(\mu(\langle y \rangle g(y, z)), \mu(\langle y \rangle g(y, x)))$$



Nous avons un radical à la position 11 dans t , avec un contexte c_1 et une valuation σ_1 :

$$\begin{aligned} c_1 &= \langle x \rangle g(\square, \mu(\langle y \rangle g(y, x))) \\ \sigma_1 &= \{Z \mapsto \underline{\lambda}x_1. g(x_1, z)\} \end{aligned}$$

Au contraire, le pré-radical à la position 12 avec le contexte c_2 et la valuation σ_2 n'est pas autorisé par la réduction faible car la variable x liée dans c_2 est libre dans $\sigma_2(Z)$:

$$\begin{aligned} c_2 &= \langle x \rangle g(\mu(\langle y \rangle g(y, z)), \square) \\ \sigma_2 &= \{Z \mapsto \underline{\lambda}x_1. g(x_1, x)\} \end{aligned}$$

Les notions de préfixe, de squelette, et d'épine qui sont liées dans le λ -calcul aux λ -abstractions peuvent être immédiatement adaptées aux abstractions des CRS, ce que nous détaillons ci-dessous.

Définition 4.3 (Préfixe). Une **préfixe** n -aire d'un terme t est un contexte n -aire c tel qu'il existe des termes t_1, \dots, t_n satisfaisant $t = c[t_1, \dots, t_n]$. Une **fonction préfixe** est une fonction prenant une abstraction $\langle x \rangle t$ en entrée et renvoyant un préfixe de $\langle x \rangle t$.

Similairement à ce qui a été défini pour le λ -calcul, le **squelette** d'une abstraction $\langle x \rangle t$ est le préfixe clos qui peut être défini soit par ablation des expressions libres maximales soit par agrégation d'épines, tandis que son épine est constituée des chemins menant aux occurrences libres de x dans t .

Définition 4.4 (Épine). La θ -**épine** d'un terme t , notée $\langle t \rangle^\theta$, est le préfixe de t défini ainsi :

$$\begin{aligned} \langle t \rangle^\theta &= \square & \theta \cap \text{fv}(t) &= \emptyset \\ \text{Sinon :} & & & \\ \langle x \rangle^\theta &= x & x &\in \theta \\ \langle \langle x \rangle t \rangle^\theta &= \langle x \rangle \langle t \rangle^{\theta \setminus \{x\}} & & \\ \langle f(t_1, \dots, t_n) \rangle^\theta &= f(\langle t_1 \rangle^\theta, \dots, \langle t_n \rangle^\theta) & & \end{aligned}$$

L'épine d'une abstraction $\langle x \rangle t$ est son préfixe $\langle t \rangle = \langle x \rangle \langle t \rangle^{\{x\}}$. Nous noterons $\langle \mathcal{S} \rangle^\theta$ l'ensemble des θ -épines des termes sur \mathcal{S} . L'épine d'un substitut $\underline{\lambda}x_1 \dots x_n.t$ est le préfixe $\langle \underline{\lambda}x_1 \dots x_n.t \rangle = \langle t \rangle^{\{x_1, \dots, x_n\}}$ de t .

Définition 4.5 (Squelette). Le θ -squelette d'un terme t , noté $\langle\langle t \rangle\rangle^\theta$, est le préfixe de t défini ainsi :

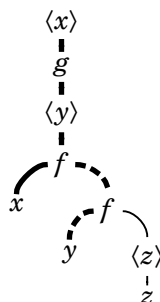
$$\begin{aligned} \langle\langle t \rangle\rangle^\theta &= \square & \theta \cap \text{fv}(t) &= \emptyset \\ \text{Sinon :} & & & \\ \langle\langle x \rangle\rangle^\theta &= x & x &\in \theta \\ \langle\langle x \rangle t \rangle^\theta &= \langle x \rangle \langle\langle t \rangle\rangle^{\theta \cup \{x\}} & & \\ \langle\langle f(t_1, \dots, t_n) \rangle\rangle^\theta &= f(\langle\langle t_1 \rangle\rangle^\theta, \dots, \langle\langle t_n \rangle\rangle^\theta) & & \end{aligned}$$

Le **squelette** d'une abstraction $\langle x \rangle t$ est son préfixe $\langle\langle t \rangle\rangle = \langle x \rangle \langle\langle t \rangle\rangle^{\{x\}}$. Nous noterons $\langle\langle \mathcal{S} \rangle\rangle^\theta$ l'ensemble des θ -squelettes des termes sur \mathcal{S} . Le **squelette** d'un substitut $\underline{\lambda}x_1 \dots x_n.t$ est le préfixe $\langle\langle \underline{\lambda}x_1 \dots x_n.t \rangle\rangle = \langle\langle t \rangle\rangle^{\{x_1, \dots, x_n\}}$ de t .

Exemple 4.6.

Considérons un symbole f d'arité 2, un symbole g d'arité 1, et le terme

$$t = \langle x \rangle g(\langle y \rangle f(x, f(y, \langle z \rangle z)))$$



L'épine du terme t est (en gras plein ci-dessus) le préfixe

$$\langle x \rangle g(\langle y \rangle f(x, \square))$$

et son squelette est (en ajoutant les pointillés)

$$\langle x \rangle g(\langle y \rangle f(x, f(y, \square)))$$

Le lemme suivant énonce pour les CRS le lemme 2.7 que nous avons pour le λ -calcul.

Lemme 4.7 (Squelettes et substitutions). Soient $\langle x \rangle t$ une abstraction et $\{y := u\}$ une substitution telles que $\langle x \rangle t^{\{y := u\}}$ est définie. Alors $\langle x \rangle t$ et $\langle x \rangle t^{\{y := u\}}$ ont même squelette.

Démonstration. Si y n'est pas libre dans $\langle x \rangle t$ alors c'est immédiat. Supposons donc $y \in \text{fv}(\langle x \rangle t)$. Les occurrences libres de y sont en particulier des expressions libres de $\langle x \rangle t$ et n'appartiennent pas au squelette. De même, la définition de $(\langle x \rangle t)^{y:=u}$ implique que les occurrences substituées de u sont libres dans $(\langle x \rangle t)^{y:=u}$ et n'influent pas sur le squelette. \square

En revanche, un certain nombre d'autres propriétés dont l'invariance des squelettes 2.8 et le respect local de la convention de Barendregt 2.14 ne sont plus vérifiées dans le cadre des CRS, ce qui va montrer tout l'intérêt du choix du système le plus robuste.

Exemple 4.8.

Considérons un symbole f d'arité 2, un symbole g d'arité 1, et la règle de réécriture

$$g(\langle x \rangle Z(x)) \rightarrow \langle x \rangle Z(f(x, x))$$

Nous avons le pas de réduction

$$g(\langle x \rangle g(x)) \rightarrow \langle x \rangle g(f(x, x))$$

dont la cible est un squelette qui n'apparaît pas dans la source, ce qui invalide une invariance des squelettes telle qu'énoncée pour le λ -calcul faible par le lemme 2.8.

Considérons maintenant la règle de réécriture

$$g(\langle x \rangle Z(x)) \rightarrow \langle x \rangle Z(Z(x))$$

Nous avons avec la valuation $\{ Z \mapsto \underline{\lambda}x_1.\langle y \rangle f(y, x_1) \}$ le pas de réduction

$$g(\langle x, y \rangle f(y, x)) \rightarrow \langle x, y \rangle f(y, \langle y \rangle f(y, x))$$

dont la cible est squelette qui ne respecte pas la convention de Barendregt, ce qui invalide la préservation du respect local de la convention Barendregt telle qu'énoncée pour le λ -calcul faible par le lemme 2.14.

4.2 Raffinements

Nous définissons dans cette section une notion de raffinement d'un ensemble de règles CRS. Intuitivement, un raffinement d'une règle $l \rightarrow r$ est un ensemble de règles qui pris dans son ensemble génère la même relation de réduction que la seule règle $l \rightarrow r$. Chaque règle du raffinement est obtenue en instanciant partiellement les méta-variables de $l \rightarrow r$, c'est-à-dire en les instanciant par des méta-termes plutôt que des termes. Nous utiliserons pour cela une notion intermédiaire de raffinement d'un motif, qui est un ensemble de valuations partielles sur lequel toute valuation sûre peut être factorisée. Pour cela, commençons par généraliser la notion de substitut en une notion de substitut partiel, qui associe des méta-termes aux méta-variables.

Définition 4.9 (Substituts partiels). *Un **substitut partiel** n -aire est une fonction notée $\underline{\lambda}x_1 \dots x_n.t$ avec t un méta-terme, qui prend n méta-termes en entrée et renvoie le méta-terme défini ainsi : $(\underline{\lambda}x_1 \dots x_n.t)(t_1, \dots, t_n) = t^{x_1:=t_1, \dots, x_n:=t_n}$.*

Exemple 4.10.

La table suivante donne à gauche plusieurs substituts partiels pour une méta-variable Z d'arité 2, et à droite l'application de ces substituts au méta-terme $\langle y, z \rangle Z(y, z)$.

$\underline{\lambda}x_1x_2.g(a)$	$\langle y, z \rangle g(a)$
$\underline{\lambda}x_1x_2.f(g(x_1), x_2)$	$\langle y, z \rangle f(g(y), z)$
$\underline{\lambda}x_1x_2.f(g(x_1), Z_1(x_2))$	$\langle y, z \rangle f(g(y), Z_1(z))$
$\underline{\lambda}x_1x_2.f(g(Z_2(x_1, x_2)), Z_3(x_1))$	$\langle y, z \rangle f(g(Z_2(y, z)), Z_3(y))$
$\underline{\lambda}x_1x_2.f(g(x_1), Z_4)$	$\langle y, z \rangle f(g(y), Z_4)$

La notion de valuation partielle est immédiatement déduite de celle de substitut partiel.

Définition 4.11. Une **valuation partielle** σ associe des substituts partiels n -aires à des méta-variables n -aires. Notons t^σ le résultat obtenu en remplaçant chaque méta-variable de t par le substitut partiel indiqué par σ . Formellement :

$$\begin{aligned} x^\sigma &= x \\ (\langle x \rangle t)^\sigma &= \langle x \rangle (t^\sigma) \\ (f(t_1, \dots, t_n))^\sigma &= f(t_1^\sigma, \dots, t_n^\sigma) \\ (Z(t_1, \dots, t_n))^\sigma &= \sigma(Z)(t_1^\sigma, \dots, t_n^\sigma) \end{aligned}$$

Une valuation partielle σ est dite **sûre** s'il n'existe pas deux substituts partiels $\sigma(Z)$ et $\sigma(Z')$ dans son codomaine et une variable x apparaissant libre dans un substitut partiel et liée dans l'autre. Une valuation partielle σ est dite **sûre pour** un méta-terme t si elle est sûre et s'il n'existe aucune méta-variable Z telle que $\sigma(Z)$ contient une occurrence libre d'une variable x qui est liée dans t .

La nouveauté est que les valuations partielles se composent.

Définition 4.12 (Composition de valuations partielles). Une valuation partielle σ sûre est sûre pour un substitut partiel $\underline{\lambda}x_1\dots x_n.t$ si $\{x_1, \dots, x_n\} \cap \text{fv}(\text{cod}(\sigma)) = \emptyset$ et si σ est sûre pour t . Étant donné un substitut $\underline{\lambda}x_1\dots x_n.t$ et une valuation partielle σ sûre pour $\underline{\lambda}x_1\dots x_n.t$, l'application de σ à $\underline{\lambda}x_1\dots x_n.t$ donne un substitut partiel défini par

$$(\underline{\lambda}x_1\dots x_n.t)^\sigma = \underline{\lambda}x_1\dots x_n.(t^\sigma)$$

Soient deux valuations partielles σ_1 et σ_2 telles que $\text{dom}(\sigma_2) \subseteq \text{mv}(\text{cod}(\sigma_1))$ et σ_2 est sûre pour tous les substituts du codomaine $\text{cod}(\sigma_1)$ de σ_1 . Leur **composition** est la valuation partielle définie par

$$\sigma_2 \circ \sigma_1 = \{Z \mapsto (\sigma_1(Z))^{\sigma_2} \mid Z \in \text{dom}(\sigma_1)\}$$

Exemple 4.13.

La composition de valuations partielles

$$\{Z \mapsto \underline{\lambda}x_1x_2.f(g(Z_2(x_1, x_2)), Z_3(x_1))\}^{\{Z_3 \mapsto \underline{\lambda}x_1.f(Z_5, f(x_1, Z_6))\}}$$

résulte en la valuation partielle

$$\{Z \mapsto \underline{\lambda}x_1x_2.f(g(Z_2(x_1, x_2)), f(Z_5, f(x_1, Z_6)))\}$$

Cette notion de composition permet de définir un ordre sur les valuations partielles reflétant la manière dont elles se factorisent.

Définition 4.14 (Factorisation de valuations partielles). *Une valuation partielle σ_f **factorise** une valuation partielle σ s'il existe une valuation partielle σ' sûre pour σ_f telle que $\sigma = \sigma' \circ \sigma_f$.*

Remarque 4.15 (Ordre de factorisation). *La factorisation définit une relation d'ordre sur les valuations partielles.*

Définition 4.16 (Raffinement d'un méta-terme). *Un **raffinement** d'un méta-terme t est un ensemble R de valuations partielles sûres pour t tel que pour toute valuation σ sûre pour t il existe une valuation partielle $\sigma_f \in R$ qui factorise σ .*

Exemple 4.17 (Raffinement exhaustif).

*Considérons un méta-terme t , et l'ensemble R des valuations sûres pour t . Alors R est un raffinement de t , appelé **raffinement exhaustif** de t .*

Exemple 4.18 (Raffinement identité).

Considérons un méta-terme t , et le singleton $\{\sigma_i\}$ où

$$\sigma_i = \{Z \mapsto \underline{\lambda}x_1 \dots x_n.Z(x_1, \dots, x_n) \mid n \in \mathbb{N}, Z \in \text{mv}(t) \text{ d'arité } n\}$$

*Remarquons que la valuation partielle σ_i est sûre pour t . De plus, le singleton $\{\sigma_i\}$ est un raffinement de t , appelé **raffinement identité**. En outre, tout ensemble R' de valuations partielles sûres pour t contenant σ est un raffinement de t , car plus généralement tout ensemble de valuations partielles sûres contenant un raffinement est toujours un raffinement.*

Exemple 4.19.

Considérons une signature $\mathcal{S} = \{f, g\}$ où les symboles f et g ont pour arité respective 2 et 1, et prenons le méta-terme $t = \langle x \rangle Z(x)$. Considérons l'ensemble de valuations partielles sûres pour t

$$R = \left\{ \begin{array}{l} \{Z \mapsto \underline{\lambda}x_1.x_1\} \\ \{Z \mapsto \underline{\lambda}x_1.\langle y \rangle Z_1(x_1, y)\} \\ \{Z \mapsto \underline{\lambda}x_1.f(Z_2(x_1), Z_3(x_1))\} \\ \{Z \mapsto \underline{\lambda}x_1.Z_5\} \end{array} \right\}$$

L'ensemble de valuations partielles R n'est pas un raffinement du méta-terme t . En effet, il existe une valuation sûre pour t

$$\{Z \mapsto \underline{\lambda}x_1.g(x_1)\}$$

qui ne se factorise pas sur R . En revanche l'ensemble de valuations partielles

$$R' = R \cup \{Z \mapsto \underline{\lambda}x_1.g(Z_4(x_1))\}$$

est un raffinement de t .

Il est également possible de remplacer la valuation $\{Z \mapsto \underline{\lambda}x_1.Z_5\}$ par l'ensemble des $\{Z \mapsto \underline{\lambda}x_1.y\}$ pour toute variable $y \in \mathcal{X} \setminus \{x\}$. Remarquons en outre qu'avec la valuation partielle $\{Z \mapsto \underline{\lambda}x_1.Z_5\}$ l'exclusion de la variable x est prise en charge par la condition de sûreté.

Définition 4.20 (Raffinement d'une règle). Soit $l \rightarrow r$ une règle CRS. Un **raffinement** de $l \rightarrow r$ est un ensemble de règles de la forme $\{l^\sigma \rightarrow r^\sigma \mid \sigma \in R\}$ avec R un raffinement de l sûr pour l et r et tel que pour toute valuation partielle $\sigma \in R$ le méta-terme l^σ est un motif.

Exemple 4.21.

Soit la signature $\mathcal{S} = \{f, g\}$ où les symboles f et g ont pour arité respective 2 et 1. Considérons la règle $d : @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0))$. Nous pouvons combiner le raffinement identité de Z_0 (exemple 4.18) avec le raffinement de Z donné par l'exemple 4.19 pour obtenir l'ensemble de valuations partielles

$$R = \left\{ \begin{array}{l} \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.x_1\} \\ \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.\langle y \rangle Z_1(x_1, y)\} \\ \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.f(Z_2(x_1), Z_3(x_1))\} \\ \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.g(Z_4(x_1))\} \\ \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.Z_5\} \end{array} \right\}$$

qui est un raffinement du motif $@(\langle x \rangle Z(x), Z_0)$ sûr pour $@(\langle x \rangle Z(x), Z_0)$ et $Z(Z(Z_0))$. Ainsi l'ensemble de règles

$$\mathcal{R}_d = \left\{ \begin{array}{l} @(\langle x \rangle x, Z_0) \rightarrow Z_0 \\ @(\langle x, y \rangle Z_1(x, y), Z_0) \rightarrow \langle y \rangle Z_1(\langle y \rangle Z_1(Z_0, y), y) \\ @(\langle x \rangle f(Z_2(x), Z_3(x)), Z_0) \rightarrow f(Z_2(f(Z_2(Z_0), Z_3(Z_0))), Z_3(f(Z_2(Z_0), Z_3(Z_0)))) \\ @(\langle x \rangle g(Z_4(x)), Z_0) \rightarrow g(Z_4(g(Z_4(Z_0)))) \\ @(\langle x \rangle Z_5, Z_0) \rightarrow Z_5 \end{array} \right\}$$

est un raffinement de la règle d .

Lemme 4.22 (Raffinement). Soit $l \rightarrow r$ une règle CRS et \mathcal{R} un raffinement de $l \rightarrow r$. Alors l'ensemble de règles \mathcal{R} génère la même relation de réduction que la seule règle $l \rightarrow r$.

Démonstration. Notons $\mathcal{R} = \{l^\sigma \rightarrow r^\sigma \mid \sigma \in R\}$ avec R un raffinement de l sûr pour l et r et tel que pour toute valuation partielle $\sigma \in R$ le méta-terme l^σ est un motif.

Soit un pas de réduction $c[l^\sigma] \rightarrow c[r^\sigma]$. Par définition de la réduction σ est une valuation sûre pour l et r . Par définition d'un raffinement il existe une valuation partielle $\sigma_1 \in R$ (sûre pour l et r) et une valuation σ_2 sûre pour l^{σ_1} et

r^{σ_1} telles que $\sigma = \sigma_2 \circ \sigma_1$. De plus la règle $l^{\sigma_1} \rightarrow r^{\sigma_1}$ est dans \mathcal{R} . Donc par sûreté de σ_2 la réduction $c[(l^{\sigma_1})^{\sigma_2}] \rightarrow c[(r^{\sigma_1})^{\sigma_2}]$ est permise par le raffinement \mathcal{R} .

Soit une réduction $c[l'^{\sigma}] \rightarrow c[r'^{\sigma}]$ avec $l' \rightarrow r' \in \mathcal{R}$. Par définition de \mathcal{R} il existe une valuation partielle $\sigma' \in R$ (sûre pour l et r) telle que $l' = l^{\sigma'}$ et $r' = r^{\sigma'}$. Par définition de la réduction σ est une valuation sûre pour l' . Donc par définition du raffinement la valuation $\sigma \circ \sigma'$ est sûre pour l et r . Donc la réduction $c[l^{\sigma \circ \sigma'}] \rightarrow c[r^{\sigma \circ \sigma'}]$ est permise par la règle $l \rightarrow r$. \square

Un raffinement R d'une règle $l \rightarrow r$ donne rarement un ensemble orthogonal de règles. En effet, des recouvrements entre règles raffinées sont susceptibles de survenir d'une part s'il existe une valuation dont la factorisation n'est pas unique, et d'autre part si le codomaine d'une valuation partielle contient un préfixe d'un membre gauche de règle. Les exemples 4.23 et 4.24 illustrent ces deux possibilités.

Exemple 4.23.

Considérons la règle $d : @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0))$, et un raffinement R de $@(\langle x \rangle Z(x), Z_0)$ sûr pour d contenant les deux valuations partielles suivantes :

$$\begin{aligned} \sigma_1 &= \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.f(g(Z_1), Z_2)\} \\ \sigma_2 &= \{Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.f(Z_3, g(Z_4))\} \end{aligned}$$

Le raffinement correspondant de d contient les deux règles

$$\begin{aligned} d_1 &: @(\langle x \rangle f(g(Z_1), Z_2), Z_0) \rightarrow f(g(Z_1), Z_2) \\ d_2 &: @(\langle x \rangle f(Z_3, g(Z_4)), Z_0) \rightarrow f(Z_3, g(Z_4)) \end{aligned}$$

qui forment un recouvrement. Le système obtenu n'est donc pas orthogonal.

Exemple 4.24.

Considérons la règle $g : g(Z) \rightarrow f(Z, a)$ et un raffinement de $g(Z)$ sûr pour g contenant la valuation partielle suivante :

$$\sigma = \{Z \mapsto \underline{\lambda}\phi.g(Z')\}$$

Nous obtenons une règle raffinée

$$g' : g(g(Z')) \rightarrow f(g(Z'), a)$$

qui forme un recouvrement avec elle-même dans le terme $g(g(g(a)))$. Le système obtenu n'est donc pas orthogonal.

Remarquons enfin que certains des recouvrements que peut introduire un raffinement ont un impact limité sur le système.

Exemple 4.25.

Reprenons la règle $g : g(Z) \rightarrow f(Z, a)$, avec un raffinement de $g(Z)$ sûr pour g et contenant les deux valuations partielles suivantes :

$$\begin{aligned} \sigma_1 &= \{Z \mapsto \underline{\lambda}\phi.\langle y \rangle g(Z_1(y))\} \\ \sigma_2 &= \{Z \mapsto \underline{\lambda}\phi.y\} \end{aligned}$$

Le raffinement correspondant de g contient les deux règles

$$\begin{aligned} g_1 & : g(\langle y \rangle g(Z_1(y))) \rightarrow f(\langle y \rangle g(Z_1(y)), a) \\ g_2 & : g(y) \rightarrow f(g(y), a) \end{aligned}$$

Ces deux règles forment un recouvrement sur le radical $g(\langle y \rangle g(y))$. Cependant le sous-terme $g(y)$, bien qu'ayant la forme d'un radical pour g_2 , ne peut en aucun cas donner lieu à un pas de réduction faible. En d'autres termes, dans le cadre de ce recouvrement l'un des deux radicaux concerné n'est pas un vrai radical. Un tel système sera dit **orthogonal en réduction faible** (voir définition 4.33).

Nous utiliserons dans la suite un raffinement particulier qualifié de *canonique*. Vis-à-vis de l'ordre de factorisation, le raffinement canonique d'une règle peut être vu à la fois comme le plus grand raffinement donnant un système orthogonal en réduction faible et comme le plus petit raffinement dont les règles n'utilisent aucune méta-variable d'arité différente de zéro. L'idée est de remplacer une méta-variable n -aire appliquée à un ensemble de variables θ par un θ -squelette dont les trous ont été remplacés par des méta-variables fraîches d'arité 0.

Définition 4.26 (Raffinement canonique d'une méta-variable). Soient \mathcal{S} une signature et Z une méta-variable d'arité n . Pour tout entier naturel k , notons $\langle\langle \mathcal{S} \rangle\rangle_k^{\{x_1, \dots, x_n\}}$ l'ensemble des $\{x_1, \dots, x_n\}$ -squelettes sur \mathcal{S} ayant k trous. Pour tout entier naturel k et pour tout squelette $s \in \langle\langle \mathcal{S} \rangle\rangle_k^{\{x_1, \dots, x_n\}}$, soient Z_1^s, \dots, Z_k^s k méta-variables fraîches d'arité 0. Le **raffinement canonique** de Z est l'ensemble de valuations partielles

$$R(Z) = \{ \{Z \mapsto \underline{\lambda}x_1 \dots x_n . (s[Z_1^s, \dots, Z_k^s])\} \mid k \in \mathbb{N}, s \in \langle\langle \mathcal{S} \rangle\rangle_k^{\{x_1, \dots, x_n\}} \}$$

Le raffinement canonique d'un méta-terme t est l'ensemble des valuations formées en choisissant un substitut partiel dans le raffinement canonique de chacune des méta-variables de t .

Définition 4.27 (Raffinement canonique d'un méta-terme). Soit t un méta-terme. Le **raffinement canonique** de t est l'ensemble de valuations partielles

$$R(t) = \prod_{Z \in \text{mv}(t)} R(Z)$$

Remarque 4.28. Toutes les valuations partielles du raffinement canonique d'un méta-terme t sont closes. En particulier elles sont sûres pour t .

Exemple 4.29.

Considérons la signature $\mathcal{S} = \{f, g, h\}$ où les symboles f , g et h ont pour arité respectivement 2, 1 et 2, et prenons le méta-terme $t = Z(Z(Z_0))$. Soient s un $\{x_1\}$ -squelette d'arité k et Z_1^s, \dots, Z_k^s k méta-variables fraîches d'arité 0. Alors la valuation partielle

$$\sigma = \{ Z_0 \mapsto \underline{\lambda}\phi . Z_0, Z \mapsto \underline{\lambda}x_1 . s[Z_1^s, \dots, Z_k^s] \}$$

avec

$$t^\sigma = s^{\{x_1 := s^{(x_1 := Z_0)}[Z_1^s, \dots, Z_k^s]\}}[Z_1^s, \dots, Z_k^s]$$

est dans $R(t)$. Remarquons au passage que le raffinement d'une méta-variable d'arité 0 est un singleton. Un exemple de valuation partielle dans $R(t)$ est

$$\sigma_1 = \{ Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.g(g(x_1)) \}$$

qui donne

$$t^{\sigma_1} = g(g(g(g(Z_0))))$$

Un autre est

$$\sigma_2 = \{ Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.Z_1 \}$$

qui donne

$$t^{\sigma_2} = Z_1$$

Un dernier est

$$\sigma_3 = \{ Z_0 \mapsto \underline{\lambda}\phi.Z_0, Z \mapsto \underline{\lambda}x_1.f(Z_2, h(x_1, Z_3)) \}$$

qui donne

$$t^{\sigma_3} = f(Z_2, h(f(Z_2, h(Z_0, Z_3)), Z_3))$$

Le raffinement canonique $R(t)$ d'un méta-terme t est un raffinement qui permet de factoriser chaque valuation sûre pour t d'une unique manière.

Lemme 4.30 (Factorisation canonique). *Considérons un méta-terme t et son raffinement canonique $R(t)$. Soit σ une valuation sûre pour t . Alors il existe une unique valuation partielle $\sigma_f \in R(t)$ telle que σ_f factorise σ .*

En particulier, le raffinement canonique $R(t)$ est un raffinement de t .

Démonstration. Soit σ une valuation sûre pour t . Soit $Z \in \text{mv}(t)$ d'arité n , et s le $\{x_1, \dots, x_n\}$ -squelette du terme $\sigma(Z)(x_1, \dots, x_n)$. Notons Z_1, \dots, Z_k les k méta-variables fraîches d'arité 0 associées à la paire (Z, s) . Nous pouvons alors définir $\sigma_f(Z) = \underline{\lambda}x_1 \dots x_n. s[Z_1, \dots, Z_k]$. Soient de plus t_1, \dots, t_k les termes tels que $s[t_1, \dots, t_k] = \sigma(Z)(x_1, \dots, x_n)$. Nous pouvons cette fois définir pour tout i $\sigma'(Z_i) = \underline{\lambda}\phi.t_i$.

Vérifions que ces substituts sont sûrs pour t^{σ_f} . Soit x une variable libre dans $\underline{\lambda}\phi.t_i$. Par définition du squelette s , l'expression t_i est une expression libre de $s[t_1, \dots, t_k]$. En particulier x est libre dans $s[t_1, \dots, t_k]$. La sûreté de σ assure qu'aucune variable libre de $s[t_1, \dots, t_k]$ n'est liée dans t , donc x n'est pas liée dans t et les substituts $\sigma'(Z_i)$ sont alors sûrs pour t^{σ_f} .

En répétant l'opération par récurrence sur le nombre de méta-variables dans t nous obtenons une valuation partielle $\sigma_f \in R(t)$ et une valuation σ' sûre pour t^{σ_f} et telle que $\sigma' \circ \sigma_f = \sigma$.

Supposons maintenant qu'existent une valuation partielle $\sigma'_f \in R(t)$ et une valuation σ'' telles que $\sigma = \sigma'' \circ \sigma'_f$ et $\sigma'_f \neq \sigma_f$. Notons $\sigma'_f(Z) = \underline{\lambda}x_1 \dots x_n. s'[Z_1^{s'}, \dots, Z_k^{s'}]$.

Par définition $s' = \langle\langle s'[\sigma''(Z_1^{s'}), \dots, \sigma''(Z_k^{s'})] \rangle\rangle^{\{x_1, \dots, x_n\}}$. Or $\sigma'' \circ \sigma'_f = \sigma = \sigma' \circ \sigma_f$, donc $s'[\sigma''(Z_1^{s'}), \dots, \sigma''(Z_k^{s'})] = s[\sigma'(Z_1^s), \dots, \sigma'(Z_k^s)]$ et $s' = \langle\langle s[\sigma'(Z_1^s), \dots, \sigma'(Z_k^s)] \rangle\rangle^{\{x_1, \dots, x_n\}} = s$. Donc $\sigma'_f(Z) = \sigma_f(Z)$ et nous avons bien unicité de la factorisation. \square

Corollaire 4.31 (Raffinement canonique). *Pour toute règle CRS $l \rightarrow r$ l'ensemble de règles $\{l^\sigma \rightarrow r^\sigma \mid \sigma \in R(l)\}$ est un raffinement de $l \rightarrow r$ appelé **raffinement canonique**.*

Exemple 4.32.

Considérons la signature $\mathcal{S} = \{f, g, h\}$ où les symboles f , g et h ont pour arité respectivement 2, 1 et 2. Le raffinement canonique de la règle

$$@(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0))$$

contient entre autres les trois règles suivantes

$$\begin{aligned} @(\langle x \rangle g(g(x)), Z_0) &\rightarrow g(g(g(Z_0))) \\ @(\langle x \rangle Z_1, Z_0) &\rightarrow Z_1 \\ @(\langle x \rangle f(Z_2, h(x, Z_3)), Z_0) &\rightarrow f(Z_2, h(f(Z_2, h(Z_0, Z_3)), Z_3)) \end{aligned}$$

Enfin, nous allons vérifier que le raffinement canonique d'un ensemble de règle n'est pas une source de non-orthogonalité en réduction faible. En effet, si le raffinement canonique d'un système crée un nouveau recouvrement entre deux radicaux $l_1^{\sigma_1}$ et $l_2^{\sigma_2}$, alors l'un de ces radicaux aura nécessairement une position gelée.

Définition 4.33 (Orthogonalité en réduction faible). *Soit $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS. Le système Σ est **sans recouvrement en réduction faible** si les deux points suivants sont vérifiés.*

1. Si un radical $l_1^{\sigma_1}$ pour une règle κ_1 contient un radical $l_2^{\sigma_2}$ pour une règle $\kappa_2 \neq \kappa_1$, alors l'un des deux points suivants est vérifié :
 - il existe une méta-variable $Z_1 \in \text{mv}(l_1)$ telle que le radical $l_2^{\sigma_2}$ est déjà contenu dans $\sigma_1(Z_1)$, ou
 - la position p à laquelle $l_2^{\sigma_2}$ apparaît dans $l_1^{\sigma_1}$ est une position gelée de $l_1^{\sigma_1}$.
2. De même si un radical pour une règle κ_1 contient strictement un autre radical pour la même règle κ_1 .

Le système Σ est **orthogonal en réduction faible** s'il est à la fois linéaire gauche et sans recouvrement en réduction faible.

Lemme 4.34 (Orthogonalité du raffinement canonique). *Soit \mathcal{R} un ensemble orthogonal de règles. Le raffinement canonique $R(\mathcal{R})$ de \mathcal{R} est orthogonal en réduction faible.*

Démonstration. Soient $\kappa : l \rightarrow r$ une règle de \mathcal{R} et $l^\sigma \rightarrow r^\sigma$ une règle de $R(\kappa)$. Supposons qu'une méta-variable Z apparaisse deux fois dans l^σ . Par orthogonalité de \mathcal{R} la règle κ est linéaire gauche, donc l contient au plus une occurrence

de Z , et il existe une méta-variable Z_0 telle que $\sigma(Z_0)$ contient au moins une occurrence de Z . De plus les méta-variables du codomaine de σ sont prises fraîches, donc Z ne peut avoir d'occurrence hors de $\sigma(Z_0)$. Par linéarité gauche de κ la méta-variable Z_0 n'apparaît qu'une fois dans l , à une position p_0 . De plus l est un motif, donc aucune position $p < p_0$ n'est la position d'une méta-variable. Alors la position p_0 n'a qu'une position résultante dans l^σ , et le substitut partiel $\sigma(Z_0)$ n'est utilisé qu'une fois. Donc $\sigma(Z_0) = \underline{\lambda}x_1\dots x_n.t$ contient deux occurrences de Z , ce qui contredit la définition du raffinement canonique. Donc les règles de $R(\mathcal{R})$ sont linéaires à gauche.

Vérifions maintenant que $R(\mathcal{R})$ est sans recouvrement en réduction faible. Dans $R(\mathcal{R})$, considérons un radical $l_o^{\sigma_o}$ pour une règle $\kappa_o : l_o \rightarrow r_o$ contenant un sous-radical $l_i^{\sigma_i}$ pour une règle $\kappa_i : l_i \rightarrow r_i$ telle que $\kappa_o \neq \kappa_i$. Notons $l_o^{\sigma_o} = c_i[l_i^{\sigma_i}]$. Par définition de $R(\mathcal{R})$, il existe deux règles $\kappa'_o : l'_o \rightarrow r'_o$ et $\kappa'_i : l'_i \rightarrow r'_i$ de \mathcal{R} et deux valuations partielles σ'_o et σ'_i telles que $l_o = (l'_o)^{\sigma'_o}$ et $l_i = (l'_i)^{\sigma'_i}$. De plus $\sigma_o \circ \sigma'_o$ (resp. $\sigma_i \circ \sigma'_i$) est sûre pour κ_o (resp. κ_i). Donc $(l'_o)^{\sigma_o \circ \sigma'_o}$ et $(l'_i)^{\sigma_i \circ \sigma'_i}$ sont deux radicaux dans \mathcal{R} et le second est toujours un sous-radical du premier. Trois cas sont à considérer en fonction de κ'_i et c_i .

- I) Si $c_i = \square$ et $\kappa'_i = \kappa'_o$, alors $(l'_o)^{\sigma_o \circ \sigma'_o} = (l'_i)^{\sigma_i \circ \sigma'_i}$ et $l'_o = l'_i$, et donc $\sigma_o \circ \sigma'_o = \sigma_i \circ \sigma'_i$. Alors par lemme 4.30 de factorisation canonique $\sigma'_o = \sigma'_i$, d'où $\kappa_o = \kappa_i$, et il ne s'agit pas d'un recouvrement (les deux radicaux définissent le même pas de réduction).
- II) Si $c_i \neq \square$ et $\kappa'_i = \kappa'_o$, alors par absence de recouvrement dans \mathcal{R} (deuxième point) nous sommes dans l'un des cas suivants :
 - Le radical $(l'_i)^{\sigma_i \circ \sigma'_i}$ est contenu dans un substitut $\underline{\lambda}x_1\dots x_n.t_o$ du codomaine de la valuation $\sigma_o \circ \sigma'_o$. Le substitut $\underline{\lambda}x_1\dots x_n.t_o$ est de la forme $\underline{\lambda}x_1\dots x_n.(t'_o)^{\sigma_o}$ où le substitut partiel $\underline{\lambda}x_1\dots x_n.t'_o$ est dans le codomaine de la valuation partielle σ'_o . Nous avons deux cas à considérer en fonction de la position p_0 du radical $(l'_i)^{\sigma_i \circ \sigma'_i}$ dans t_o .
 - Si p_0 est une position de t'_o , alors par définition du raffinement canonique p_0 est une position dynamique de t_o , c'est-à-dire que p_0 correspond à une position gelée de $(l'_o)^{\sigma_o \circ \sigma'_o}$.
 - Sinon $(l'_i)^{\sigma_i \circ \sigma'_i}$ est contenu dans le codomaine de la valuation σ_o .
 - La position p du radical $(l'_i)^{\sigma_i \circ \sigma'_i}$ est une position gelée de $(l'_o)^{\sigma_o \circ \sigma'_o}$.
- III) De même si $\kappa'_i \neq \kappa'_o$, indépendamment de c_i , avec le premier point de la définition de l'absence de recouvrement en réduction faible.

De même encore si $\kappa_o = \kappa_i$ et $c_i \neq \square$, et finalement $R(\mathcal{R})$ est sans recouvrement en réduction faible. Donc $R(\mathcal{R})$ est orthogonal en réduction faible. \square

Cette notion d'orthogonalité en réduction faible est plus faible que la notion d'orthogonalité de référence (définition 0.77). Cependant, les systèmes orthogonaux en réduction faible se comporteront pour ce qui nous intéresse comme des systèmes orthogonaux, et nous verrons au chapitre 9 que le raffinement canonique d'un système orthogonal, qui n'est qu'orthogonal en réduction faible, peut être transformé en système orthogonal.

4.3 Définition du partage pleinement paresseux

Nous allons maintenant pouvoir procéder à l'étiquetage des *CRS*. Précisons que la discussion précédente sur l'orthogonalité n'était qu'une parenthèse : le raffinement canonique et la construction que nous allons maintenant définir s'appliquent à tout *CRS*, même non-orthogonal.

Une fois acquis le fait que nous nous concentrerons sur la notion de squelette qui était au cœur du système \mathcal{P}_W du chapitre 2, il reste à déterminer les autres caractéristiques du λ -calcul étiqueté du chapitre 2 que nous allons conserver. Rappelons-en deux :

- Au chapitre 2 toute étiquette créée par un pas de réduction est de la forme $[\omega, \alpha]$ où ω est l'étiquette racine du β -radical et α est l'étiquette de la position ancêtre.
- Au chapitre 2 la vérification de l'axiome de partage est simplifiée par le fait que toutes les étiquettes de la zone d'effet d'un pas de réduction sont différentes les unes des autres.

Combiner ces deux caractéristiques dans le cadre général des *CRS* est assez technique. La seconde caractéristique en revanche s'articule très naturellement avec le raffinement canonique d'un système et sera donc retrouvée ici. Une méthode alternative reprenant uniquement la première caractéristique sera explorée au chapitre 11.

L'idée directrice de l'étiquetage de cette section est la suivante : considérons un pas de réduction ρ dont le radical a pour étiquette racine ω , alors toute position propre p du réduit de ρ se verra attribuer l'étiquette $[\omega, p]$. Nous nous assurons ainsi que toutes les étiquettes de la zone d'effet de ρ sont différentes les unes des autres.

Définition 4.35 (Étiquettes). *L'ensemble d'étiquettes \mathcal{L} est défini par la grammaire suivante :*

$$\alpha ::= p \mid [\alpha, p] \quad p \text{ position}$$

Définition 4.36 (Signature étiquetée). *Soit \mathcal{S} une signature CRS. La **signature étiquetée** $\mathcal{S}^{\mathcal{L}}$ est définie par*

$$\mathcal{S}^{\mathcal{L}} = \bigcup_n \mathcal{S}_n^{\mathcal{L}}$$

avec pour toute arité n

$$\mathcal{S}_n^{\mathcal{L}} = \{f^\alpha \mid f \in \mathcal{S}_n, \alpha \in \mathcal{L}\}$$

Nous notons $(.)^\bullet : \mathcal{S}^{\mathcal{L}} \rightarrow \mathcal{S}$ la fonction d'oubli des étiquettes telle que pour tout symbole $f \in \mathcal{S}$ et toute étiquette $\alpha \in \mathcal{L}$

$$(f^\alpha)^\bullet = f$$

Remarque 4.37. *L'utilisation systématique par convention 0.57 d'un symbole \diamond pour marquer les lieurs se révèle pratique à ce point, puisqu'il permet par le même*

procédé de donner des étiquettes non seulement aux symboles de la signature d'origine, mais également aux lieux, tout en restant dans le cadre usuel des CRS. En effet, des symboles \diamond^α sont automatiquement présents dans la signature étiquetée $\mathcal{S}^\mathcal{L}$. Pour toute étiquette α , nous noterons $\langle x \rangle^\alpha t$ pour abrégier $\diamond^\alpha([x]t)$.

Définition 4.38 (Étiquetage d'un méta-terme). Soit \mathcal{S} une signature CRS. La fonction de désétiquetage $(.)^\bullet$ induit un morphisme des méta-termes sur $\mathcal{S}^\mathcal{L}$ vers les méta-termes sur \mathcal{S} . Un méta-terme t^l sur $\mathcal{S}^\mathcal{L}$ est un **étiquetage** d'un méta-terme t sur \mathcal{S} si $(t^l)^\bullet = t$.

À tout méta-terme t sur une signature \mathcal{S} peut être associé un méta-terme t^l sur $\mathcal{S}^\mathcal{L}$ qualifié d'*initial*. Le méta-terme t^l est obtenu en remplaçant tout symbole f de \mathcal{S} à la position p par le symbole f^p de $\mathcal{S}^\mathcal{L}$.

Définition 4.39 (Étiquetage initial). Soit \mathcal{S} une signature CRS. Pour toute position p la fonction $i_p(.)$ transforme un terme sur \mathcal{S} en un terme sur $\mathcal{S}^\mathcal{L}$ par les règles suivantes :

$$\begin{aligned} i_p(x) &= x \\ i_p([x]t) &= [x]i_p(t) \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p-1}(t_1), \dots, i_{p-n}(t_n)) \end{aligned}$$

Soit t un méta-terme sur une signature \mathcal{S} . L'**étiquetage initial** de t est le méta-terme $i_c(t)$, que nous notons encore $i(t)$.

Rappelons que par convention nous avons

$$i_p(\langle x \rangle t) = i_p(\diamond([x]t)) = \diamond^p(i_{p-1}([x]t)) = \diamond^p([x]i_{p-1}(t)) = \langle x \rangle^p i_{p-1}(t)$$

Exemple 4.40.

$$\begin{aligned} i(Z(\mu(\langle x \rangle Z(x)))) &= Z(\mu^1(\langle x \rangle^{11}(Z(x)))) \\ i(g(g(g(Z_0)))) &= g^\epsilon(g^1(g^{11}(g^{111}(Z_0)))) \\ i(f(Z_2, h(f(Z_2, h(Z_0, Z_3)), Z_3))) &= f^\epsilon(Z_2, h^2(f^{21}(Z_2, h^{212}(Z_0, Z_3)), Z_3)) \end{aligned}$$

L'opération de réétiquetage uniforme sur les λ -termes étiquetés (définition 2.18) est immédiatement étendue aux termes CRS sur une signature étiquetée.

Définition 4.41 (Réétiquetage uniforme). Pour tout terme t sur une signature $\mathcal{S}^\mathcal{L}$ et étiquette $\omega \in \mathcal{L}$, le réétiquetage uniforme $[\omega, t]$ est défini ainsi :

$$\begin{aligned} [\omega, x] &= x \\ [\omega, [x]t] &= [x][\omega, t] \\ [\omega, f^\alpha(t_1, \dots, t_n)] &= f^{[\omega, \alpha]}([\omega, t_1], \dots, [\omega, t_n]) \end{aligned}$$

Les étiquettes ne doivent pas interférer avec la réduction, et en particulier elles ne doivent en aucun cas empêcher l'existence d'un pas de réduction étiqueté là où existe un pas de réduction non étiqueté. Par conséquent l'étiquetage d'une règle contient une infinité de règles correspondant à tous les étiquetages possibles du membre gauche. Nous utiliserons en revanche une seule manière d'étiqueter le membre droit, qui sera déduite de son étiquetage initial.

Définition 4.42 (Règle étiquetée). Soit $\kappa : l \rightarrow r$ une règle CRS. L'ensemble $\kappa^{\mathcal{L}}$ des **étiquetages de la règle** $\kappa : l \rightarrow r$ est l'ensemble de règles étiquetées

$$\kappa^{\mathcal{L}} = \{ l_\alpha \rightarrow [\tau(l_\alpha), i(r)] \mid l_\alpha^\bullet = l \}$$

Si \mathcal{R} est un ensemble de règles CRS, alors nous notons $\mathcal{R}^{\mathcal{L}}$ l'ensemble des étiquetages des règles de \mathcal{R} .

Exemple 4.43.

Considérons la règle CRS

$$@(\langle x \rangle f(Z_1, g(x))) \rightarrow f(Z_1, g(f(Z_1, g(Z_0))))$$

Trois de ses étiquetages sont montrés ci-dessous. Remarquons en particulier que seule l'étiquette racine du membre gauche d'une règle apparaît dans la construction des étiquettes de son membre droit, et que tout étiquetage même incohérent du membre gauche est accepté.

$$\begin{aligned} @^\omega(\langle x \rangle^\alpha f^\beta(Z_1, g^\gamma(x))) &\rightarrow f^{[\omega, \epsilon]}(Z_1, g^{[\omega, 2]}(f^{[\omega, 21]}(Z_1, g^{[\omega, 212]}(Z_0)))) \\ @^\omega(\langle x \rangle^\beta f^\beta(Z_1, g^\alpha(x))) &\rightarrow f^{[\omega, \epsilon]}(Z_1, g^{[\omega, 2]}(f^{[\omega, 21]}(Z_1, g^{[\omega, 212]}(Z_0)))) \\ @^\alpha(\langle x \rangle^\gamma f^\omega(Z_1, g^\beta(x))) &\rightarrow f^{[\alpha, \epsilon]}(Z_1, g^{[\alpha, 2]}(f^{[\alpha, 21]}(Z_1, g^{[\alpha, 212]}(Z_0)))) \end{aligned}$$

Ainsi, de tout CRS $(\mathcal{S}, \mathcal{R})$ nous pouvons déduire un CRS $(\mathcal{S}^{\mathcal{L}}, \mathcal{R}^{\mathcal{L}})$ manipulant des termes étiquetés. Nous pouvons de plus vérifier que si le CRS de départ est le raffinement canonique d'un CRS faible, alors le système étiqueté déduit est un SLS. Pour cela nous aurons également l'utilité du lemme suivant justifiant l'appellation de la notion de positions statiques.

Lemme 4.44 (Positions statiques). Soit $\rho : t = c[l^\sigma] \xrightarrow{q} c[r^\sigma] = t'$ une réduction, et p' une position statique de t' . Alors p' a un ancêtre p dans t , et $t|_p = t'|_{p'}$.

Démonstration. La position p' , en tant que position statique, se décompose en $p' = q \cdot p'_Z \cdot p_\sigma$, où p'_Z est une position résultante d'une position d'une méta-variable Z dans r , et p_σ est une position statique de $\sigma(Z)$. Par définition d'une règle de réduction, la méta-variable Z de r apparaît aussi dans l . Soit p_Z une position de la méta-variable Z dans l . Alors la position $p = q \cdot p_Z \cdot p_\sigma$ est un ancêtre de la position $p' = q \cdot p'_Z \cdot p_\sigma$, et finalement

$$t|_p = l^\sigma|_{p_Z \cdot p_\sigma} = \sigma(Z)(t_1, \dots, t_n)|_{p_\sigma} = \sigma(Z)|_{p_\sigma} = \sigma(Z)(t'_1, \dots, t'_n)|_{p_\sigma} = r^\sigma|_{p'_Z \cdot p_\sigma} = t'|_{p'}$$

□

Nous allons maintenant montrer que cet étiquetage décrit SLS pour tout CRS faible.

Définition 4.45 (Ordre sur les étiquettes). L'ensemble \mathcal{L} d'étiquettes de la définition 4.35 est muni de l'ordre partiel \leq généré par :

$$\omega \leq [\omega, \alpha]$$

Instanciation 4.46. Soit un CRS faible $(\mathcal{S}, \mathcal{R})$. Considérons $(\mathcal{S}^{\mathcal{L}}, R(\mathcal{R})^{\mathcal{L}})$ l'étiquetage de son raffinement canonique et définissons :

- \mathcal{T} l'ensemble des termes syntaxiques sur $\mathcal{X} \cup \{\langle x \rangle \mid x \in \mathcal{X}\} \cup \mathcal{S}^{\mathcal{L}}$.
- \mathcal{R} l'ensemble des pas de réduction faible par les règles de $R(\mathcal{R})^{\mathcal{L}}$.

Soit $\rho \in \mathcal{R}$. Notons $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et définissons :

- $\text{dex}(\rho) = l^\sigma$
- $\text{duct}(\rho) = r^\sigma$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \mathcal{P}_p(r, \sigma)$ (où l'ensemble des positions propres $\mathcal{P}_p(r, \sigma)$ est donné par la définition 0.70)

Notons $\Sigma = (\mathcal{S}[\mathcal{L}] \cup \mathcal{X}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$.

Lemme 4.47 (ATRS marqué). *Le système Σ est un ATRS marqué.*

Démonstration.

Source & Cible Tout terme syntaxique est dans \mathcal{T} .

Résidus Soient deux pas de réduction $\rho_1, \rho_2 \in \mathcal{R}$ tels que $\text{src}(\rho_1) = \text{src}(\rho_2)$ et les positions racines $\text{root}(\rho_1)$ et $\text{root}(\rho_2)$ sont disjointes. Notons $\rho_1 : c_1[l_1^{\sigma_1}] \rightarrow c_1[r_1^{\sigma_1}]$ et $\rho_2 : c_2[l_2^{\sigma_2}] \rightarrow c_2[r_2^{\sigma_2}]$. Comme $\text{root}(\rho_1)$ et $\text{root}(\rho_2)$ sont disjointes, il existe c un contexte binaire tel que $\text{src}(\rho_1) = \text{src}(\rho_2) = c[l_1^{\sigma_1}, l_2^{\sigma_2}]$ et $\text{tgt}(\rho_1) = c[r_1^{\sigma_1}, l_2^{\sigma_2}]$. Alors $\rho'_2 : \text{tgt}(\rho_1) \rightarrow c[r_1^{\sigma_1}, r_2^{\sigma_2}]$ est un pas de réduction de \mathcal{R} équivalent à ρ_2 .

Zone d'effet Soient $\rho \in \mathcal{R}$ et $q \in \text{pos}(\text{duct}(\rho)) \setminus \text{effz}(\rho)$. Par définition de effz , $q \in \mathcal{P}_s(r, \sigma)$. Conclusion par lemme 4.44. \square

Lemme 4.48 (SLS). *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine Par définition de \mathcal{T} , tous les symboles sauf les variables sont étiquetées, et le symbole de tête d'un radical ne peut être une variable.

Progression Soient $\rho \in \mathcal{R}$ et $\alpha \in \mathcal{L}$ telles que α apparaît dans $\text{effz}(\rho)$. L'étiquette α est de la forme $[\tau(\rho), p]$ donc $\tau(\rho) < \alpha$.

Héritage Soient $\rho \in \mathcal{R}$ et $\alpha, \beta \in \mathcal{L}$ telles que β apparaît dans $\text{effz}(\rho)$, $\alpha < \beta$ et $\alpha \neq \tau(\rho)$. En particulier β est de la forme $[\tau(\rho), p]$, donc $\alpha \leq \tau(\rho)$ et même $\alpha < \tau(\rho)$. De plus $\tau(\rho)$ apparaît dans $\text{dex}(\rho)$.

Partage Toutes les étiquettes de la zone d'effet d'un pas de réduction ρ sont de la forme $[\tau(\rho)]$ et sont différentes par définition de l'étiquetage initial $i(\cdot)$. \square

4.4 Comparaison avec le λ -calcul étiqueté

Cette section montre comment le partage pleinement paresseux défini dans ce chapitre pour les CRS généralise le partage pleinement paresseux de Wadsworth pour le λ -calcul vu au chapitre 2.

Pour cela, la relation entre le λ -calcul étiqueté du chapitre 2 et l'étiquetage du présent chapitre appliqué au λ -calcul vu comme un *CRS* est synthétisée en trois points :

- Ils sont deux étiquetages du même *ATRS* sous-jacent.
- Ils sont différents.
- Ils définissent des relations de réduction parallèle étiquetée bisimilaires.

La conjonction des premier et troisième points donne l'équivalence de ces deux notions de partage et justifie l'attribution du qualificatif « pleinement paresseux » au partage spécifié par le système défini dans le présent chapitre.

Il a déjà été mentionné que le λ -calcul pouvait être représenté par un *CRS* dont la signature \mathcal{S} contient le seul symbole binaire @ en plus du symbole distingué \diamond associé aux lieux et dont l'unique règle est

$$\beta : @(\langle x \rangle Z(x), Z') \rightarrow Z(Z')$$

En identifiant pour toute variable x les symboles λx et $\langle x \rangle$ ce λ -calcul encodé dans les *CRS* faibles définit le même *ATRS* que le λ -calcul faible.

Appliquons maintenant l'étiquetage pleinement paresseux des *CRS* faibles à cette représentation du λ -calcul. La première étape consiste à calculer le raffinement canonique de la règle β . Ce raffinement $R(\beta)$ est formé de l'union pour tout entier k et tout $\{x\}$ -squelette s avec k trous des règles

$$@(\langle x \rangle s[Z_1^s, \dots, Z_k^s], Z') \rightarrow s^{\{x:=Z'\}}[Z_1^s, \dots, Z_k^s]$$

où les k méta-variables distinctes Z_1^s, \dots, Z_k^s sont caractéristiques de la paire (k, s) .

Exemple 4.49.

Trois des règles du raffinement canonique de β sont :

$$\begin{array}{lll} @(\langle x \rangle Z_1, Z') \rightarrow Z_1 & k = 1 & s = \square \\ @(\langle x \rangle x, Z') \rightarrow Z' & k = 0 & s = x \\ @(\langle x, y \rangle @(\langle x \rangle @(\langle y \rangle Z_2, y)), Z') \rightarrow \langle y \rangle @(\langle x \rangle @(\langle y \rangle @(\square, y))) & k = 1 & s = \langle y \rangle @(\langle x \rangle @(\square, y)) \end{array}$$

Nous en déduisons un ensemble de règles étiquetées qui est l'union pour toutes étiquettes $\alpha, \omega \in \mathcal{L}$, pour tout entier k , pour tout $\{x\}$ -squelette s , et pour tout étiquetage s_l de s des règles

$$@^\omega(\langle x \rangle^\alpha s_l[Z_1^s, \dots, Z_k^s], Z') \rightarrow [\omega, i(s)]^{\{x:=Z'\}}[Z_1^s, \dots, Z_k^s]$$

Exemple 4.50.

Trois des règles étiquetées du raffinement canonique de β sont :

$$\begin{array}{lll} @^\omega(\langle x \rangle^\alpha Z_1, Z') \rightarrow Z_1 & & s_l = \square \\ @^\omega(\langle x \rangle^\alpha x, Z') \rightarrow Z' & & s_l = x \\ @^\omega(\langle x \rangle^\alpha \langle y \rangle^\beta @^\gamma(x, @^\delta(Z_2, y)), Z') \rightarrow \langle y \rangle^{[\omega, \epsilon]} @^{[\omega, 1]}(Z', @^{[\omega, 12]}(Z_2, y)) & & s_l = \langle y \rangle^\beta @^\gamma(x, @^\delta(\square, y)) \end{array}$$

Le système obtenu est différent de l'étiquetage direct du λ -calcul défini au chapitre 2. En effet, observons dans ces deux cas les nouvelles étiquettes introduites par un pas de réduction : elles ont toutes la forme $[\omega, \alpha]$ où ω est l'étiquette de l'application du radical et où α est une autre étiquette, mais l'origine de cet α diffère.

- Au chapitre 2, α est une étiquette extraite du radical.
- Dans le présent chapitre, α est une position propre du réduct.

En revanche les deux systèmes d'étiquetage ont deux propriétés communes importantes :

- Leurs pas de réduction ont les mêmes zones d'effet.
- Les étiquettes de la zone d'effet de tout pas de réduction sont toutes différentes.

Exemple 4.51.

Avec l'étiquetage du chapitre 2 nous avons le pas de réduction

$$@^\omega(\lambda^\alpha x. \lambda^\beta y. @^\gamma(x, @^\delta(\lambda^\zeta z.z, y)), \lambda^\eta z.z) \rightarrow \lambda^{[\omega, \beta]} y. @^{[\omega, \gamma]}(\lambda^\eta z.z, @^{[\omega, \delta]}(\lambda^\zeta z.z, y))$$

Et le pas de réduction équivalent avec l'étiquetage des CRS est

$$@^\omega(\langle x \rangle^\alpha \langle y \rangle^\beta @^\gamma(x, @^\delta(\langle z \rangle^\zeta z, y)), \langle z \rangle^\eta z) \rightarrow \langle y \rangle^{[\omega, \epsilon]} @^{[\omega, 1]}(\langle z \rangle^\eta z, @^{[\omega, 12]}(\langle z \rangle^\zeta z, y))$$

Les deux pas de réduction définissent la même zone d'effet $\{\epsilon, 1, 12\}$, correspondant aux squelettes non étiquetés

$$\lambda y. \square(\square y) \quad | \quad \langle y \rangle @(\square, @(\square, y))$$

Dans le premier cas les trois étiquettes différentes $[\omega, \beta]$, $[\omega, \gamma]$ et $[\omega, \delta]$ sont associées à ces positions. Dans le deuxième il s'agit de $[\omega, \epsilon]$, $[\omega, 1]$ et $[\omega, 12]$, qui de même sont différentes entre elles.

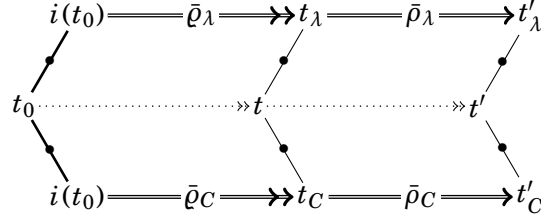
Ces propriétés communes nous permettent d'exprimer que ces deux étiquetages d'un même *ATRS* définissent le même partage. Ceci est formalisé au moyen de la notion de termes miroirs introduite à la section 1.5.

Théorème 4.52 (Partage équivalent). *Soient t_λ un terme étiqueté selon le chapitre 2 et t_C un terme étiqueté selon le présent chapitre tels que $t_\lambda \approx t_C$. Alors t_λ et t_C induisent la même relation d'équivalence par étiquettes sur les positions de $t = t_\lambda^\bullet = t_C^\bullet$.*

Démonstration. Par définition de $t_\lambda \approx t_C$, il existe un λ -terme non étiqueté t_0 et une séquence de réduction miroir parallèle $(\bar{\rho}_\lambda, \bar{\rho}_C)$ de source $(i(t_0), i(t_0))$ et de cible (t_λ, t_C) . Raisonnons par récurrence sur le nombre d'étapes de réduction parallèle étiquetée dans la séquence.

Le cas de base, avec une séquence de longueur 0, est immédiat car alors $t_\lambda = i(t_0) = t_C$.

Supposons donc que la propriété est vérifiée pour les séquences de réduction miroir parallèle de longueur n , et considérons une séquence de réduction miroir parallèle $(\bar{\rho}_\lambda \bar{\rho}'_\lambda, \bar{\rho}_C \bar{\rho}'_C)$ de cible (t'_λ, t'_C) , avec $(\bar{\rho}_\lambda, \bar{\rho}_C)$ une séquence de réduction miroir parallèle de longueur n . Notons t_λ et t_C les cibles respectives de $\bar{\rho}_\lambda$ et $\bar{\rho}_C$.



Par hypothèse de récurrence les termes t_λ et t_C induisent la même relation d'équivalence par étiquettes sur les positions de $t = t_\lambda^\bullet = t_C^\bullet$. De plus, t_λ et t_C vérifient tous deux les propriétés \mathbb{S} et \mathbb{I} par lemme 1.18 et théorème 1.19 et t_λ vérifie la propriété \mathbb{F} par lemme 2.24. Notons ω_λ et ω_C les étiquettes respectives des radicaux réduits en parallèle par $\bar{\rho}_\lambda$ et $\bar{\rho}_C$.

Soient p' et q' deux positions de t' . Notons p et q leurs ancêtres respectifs dans t . Supposons que $\tau_{p'}(t'_\lambda) = \tau_{q'}(t'_\lambda) = \alpha$. Nous allons montrer que $\tau_{p'}(t'_C) = \tau_{q'}(t'_C)$. Raisonnons par cas sur les positions p' et q' .

- Si ni p' ni q' n'est dans la zone d'effet d'un des pas de réduction parallèle, alors $\tau_{p'}(t'_C) = \tau_p(t_C) = \tau_q(t_C) = \tau_{q'}(t'_C)$.
- Si p' est dans la zone d'effet d'un des pas de réduction parallèles et pas q' , alors $\alpha = \tau_{p'}(t'_\lambda) = [\omega_\lambda, \tau_p(t_\lambda)]$ et $\alpha = \tau_{q'}(t'_\lambda) = \tau_q(t_\lambda)$, donc $\omega_\lambda \leq \tau_q(t_\lambda)$, ce qui contredit $\mathbb{I}(t_\lambda)$.
- Si p' et q' sont tous deux dans la zone d'effet d'un des pas de réduction parallèle, alors $\alpha = \tau_{p'}(t'_\lambda) = [\omega_\lambda, \tau_p(t_\lambda)]$ et $\alpha = \tau_{q'}(t'_\lambda) = [\omega_\lambda, \tau_q(t_\lambda)]$, avec en particulier $\tau_p(t_\lambda) = \tau_q(t_\lambda)$. Notons $p' = p_r \cdot p_e$ (resp. $q' = q_r \cdot q_e$) avec p_r (resp. q_r) la position du radical et p_e (resp. q_e) une position de la zone d'effet. Alors $p = p_r \cdot 11 \cdot p_e$ (resp. $q = q_r \cdot 11 \cdot q_e$) et les positions p et q sont des positions de deux occurrences d'un même squelette. Or le terme t_λ vérifie la propriété \mathbb{F} et $\tau_p(t_\lambda) = \tau_q(t_\lambda)$, donc $p_e = q_e$ et $\tau_{p'}(t'_C) = [\omega_C, p_e] = [\omega_C, q_e] = \tau_{q'}(t'_C)$.

Nous montrons similairement que $\tau_{p'}(t'_C) = \tau_{q'}(t'_C)$ implique $\tau_{p'}(t'_\lambda) = \tau_{q'}(t'_\lambda)$, ce qui conclut la preuve. \square

Ainsi l'étiquetage des CRS proposé dans ce chapitre spécifie une notion de réduction partagée qui généralise bien le partage pleinement paresseux défini par Wadsworth pour le λ -calcul [Wad71]. Nous pourrions donc légitimement utiliser les définitions de ce chapitre pour raisonner sur le partage pleinement paresseux dans le cadre général des CRS.

4.5 Discussion : complexité

Dans l'introduction nous avons cité parmi les critères de qualité d'un modèle abstrait qu'il devait, sur l'aspect étudié, refléter fidèlement le comportement des systèmes concrets modélisés. Ici, le modèle abstrait est la réduction partagée pleinement paresseuse des CRS faibles, le système concret est un programme fonctionnel compilé pour une machine raisonnable non précisée, et l'aspect étudié est le temps d'exécution. Ainsi nous devons nous assurer que les séquences de

réduction parallèles d'un CRS représentant un programme sont en relation avec le temps d'exécution de ce programme.

Dans cette section nous faisons un premier pas dans cette direction, en isolant un fragment des CRS faibles pour lequel il existe effectivement un rapport polynomial entre la longueur des séquences de réduction parallèles et le nombre de transitions effectuées par une machine de Turing pour exécuter ces réductions (nombre appelé *coût Turing* dans la suite).

Nous allons procéder en deux étapes : la section 4.5.1 introduit d'abord une mesure de coût *polynomialement invariante* (c'est-à-dire polynomialement liée au coût Turing) pour la réduction partagée pleinement paresseuse dans l'ensemble des CRS. En s'inspirant d'une mesure définie par Dal Lago et Martini pour la réduction du λ -calcul [LM06], cette mesure prend en compte la croissance des graphes représentés par les CRS étiquetés. La section 4.5.2 caractérise ensuite le fragment des CRS *plats*, dans lesquels nous pourrions montrer que la croissance des graphes est suffisamment limitée pour que la *complexité dérivationnelle* (c'est-à-dire la longueur des séquences de réduction) soit elle-même une mesure polynomialement invariante. Dans les deux cas la clé est la suivante : la croissance du graphe lors d'un pas de réduction de graphes est bornée par le nombre d'étiquettes créées dans la réduction parallèle étiquetée correspondante.

4.5.1 Une mesure de croissance

Dal Lago et Martini définissent une mesure polynomialement invariante pour les séquences de β -réduction en enregistrant la croissance des λ -termes au cours de la réduction [LM06]. Nous adaptons cette idée ici en enregistrant non la croissance des termes étiquetés, mais la croissance des graphes représentés par ces termes étiquetés.

Définition 4.53 (Taille graphique d'un terme étiqueté). *Soient \mathcal{S} une signature de CRS et t un terme étiqueté sur \mathcal{S} . La **taille graphique** $|t|^{\mathcal{G}}$ de t est le nombre d'étiquettes différentes présentes dans t .*

Quand la propriété $\mathbb{S}(t)$ est vérifiée, $|t|^{\mathcal{G}}$ donne la taille du graphe \mathcal{G}_t .

Définition 4.54 (Mesure de coût). *Le **coût** $c(\rho)$ d'une étape de réduction $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_{t'}$ est défini par $c(\rho) = \max(1, |t'|^{\mathcal{G}} - |t|^{\mathcal{G}})$. Le **coût** $c(\rho_1 \dots \rho_n)$ d'une séquence de réduction $\rho_1 \dots \rho_n$ est la somme $\sum_i c(\rho_i)$ des coûts de ses étapes successives.*

En d'autres termes si le nombre d'étiquettes augmente, ce qui correspond à une croissance du graphe représenté, alors la mesure de coût enregistre cette augmentation. Sinon le coût d'une étape est 1.

Lemme 4.55 (Croissance). *Soit $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_{t'}$ une séquence de réduction de graphe. Alors $|t'|^{\mathcal{G}} \leq |t|^{\mathcal{G}} + c(\rho)$.*

Démonstration. Par induction sur la longueur de la séquence ρ . □

En combinant les algorithmes de Wadsworth [Wad71] et Kahl [Kah99], nous savons que le coût Turing de l'application d'une règle de réduction $l \rightarrow r$ dans un graphe \mathcal{G} est polynomiale en la taille de \mathcal{G} .

Lemme 4.56 (Coût polynomial). *Le coût Turing d'une séquence de réduction faible $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_t'$ est polynomial en $|t|^{\mathcal{G}}$ et $c(\rho)$.*

Démonstration. Soit $\rho = \rho_1 \dots \rho_n$ une séquence de réduction de graphe de source \mathcal{G}_t . Par lemme 4.55 la taille $|t_i|^{\mathcal{G}}$ du graphe obtenu après l'étape $i - 1$ est inférieure à $|t|^{\mathcal{G}} + c(\rho_1 \dots \rho_i)$. La mesure étant croissante, nous avons même $|t_i|^{\mathcal{G}} \leq |t|^{\mathcal{G}} + c(\rho)$, et il existe un polynôme \mathcal{P} tel que pour tout i , le coût Turing de ρ_i est borné par $\mathcal{P}(|t|^{\mathcal{G}}, c(\rho))$. Le coût total de la séquence est donc borné par $n \times \mathcal{P}(|t|^{\mathcal{G}}, c(\rho))$. Or, pour toute étape de réduction ρ nous avons $c(\rho) \geq 1$. Donc $c(\rho) \geq n$, et le coût total de la séquence est finalement borné par $c(\rho) \times \mathcal{P}(|t|^{\mathcal{G}}, c(\rho))$. \square

Ainsi, nous avons une borne sur le coût réel de l'évaluation partagée pleinement paresseuse qui est polynomiale d'une part en la longueur de la séquence de réduction parallèle étiquetée dans les CRS (c'est-à-dire justement l'information sur laquelle se concentre notre modèle) et d'autre part en la taille des graphes représentant les étapes intermédiaires de l'évaluation partagée (c'est-à-dire cette fois quelque chose que nous ne considérons pas comme un élément important du modèle). La section suivante identifie une classe de CRS dans laquelle ce deuxième paramètre peut être ignoré.

4.5.2 Complexité dérivationnelle

Pour les systèmes du premier ordre, il n'est pas nécessaire comme dans la section précédente de considérer la taille de chaque étape intermédiaire pour obtenir une mesure du coût d'une séquence de réduction de graphes. En effet, Avanzini et Moser ont montré que dans ce cas les tailles des étapes intermédiaires sont liées polynomialement à la taille du terme d'origine et à la longueur de la séquence de réduction, et qu'il suffit donc de compter le nombre d'étapes de réduction pour avoir une mesure polynomialement invariante [AM10]. Nous démontrons ici que cette même *complexité dérivationnelle* est encore une mesure polynomialement invariante pour la réduction faible dans une certaine classe de CRS faibles dans laquelle nous pouvons borner polynomialement la croissance des graphes.

La croissance de la taille graphique d'un terme étiqueté est liée à la création de nouvelles étiquettes. Ces créations ont lieu au niveau des positions propres du membre droit d'un raffinement canonique d'une règle CRS. Ledit membre droit d'un raffinement canonique d'une règle $l \rightarrow r$ est de la forme r^σ (avec σ une valuation partielle venant du raffinement canonique du méta-terme l). Nous avons donc besoin d'estimer le nombre de symboles dans le membre droit r^σ , et notre première étape est le lemme 4.60 qui borne la *taille* de r^σ en fonction des tailles de r et σ .

Définition 4.57 (Taille d'un terme). *La **taille** d'un méta-terme t , notée $|t|$, est le nombre de positions dans t . La **taille** d'une valuation σ prend un maximum sur son codomaine : $|\sigma| = \max\{|u| \mid \lambda\theta.u \in \text{cod}(\sigma)\}$.*

Exemple 4.58.

Le méta-terme $\langle x \rangle f(Z(x), a)$ a pour taille 5, tandis que la valuation $\{ Z \mapsto \lambda x_1. f(x_1, x_1) \}$ a pour taille 3.

Lemme 4.59 (Taille d'une substitution). Soit t un méta-terme et $\{x := u\}$ une substitution telle que le méta-terme $t^{\{x := u\}}$ est défini. Alors

$$|t^{\{x := u\}}| = |t| + |\text{fo}_x(t)|(|u| - 1)$$

où $|\text{fo}_x(t)|$ désigne le cardinal de l'ensemble de positions $\text{fo}_x(t)$.

Démonstration. Par induction sur t . □

Ce lemme s'étend naturellement à un substitution n -aire.

Lemme 4.60 (Taille d'une valuation). Soit t un méta-terme et σ une valuation partielle sûre pour t . Alors $|t^\sigma| \leq (|t| + |\sigma|)^{|t|}$.

Démonstration. Par induction sur t .

- Cas x . Comme $x^\sigma = x$ nous avons $|x^\sigma| = 1$, d'où $|x^\sigma| \leq 1 + |\sigma| = (|x| + |\sigma|)^{|x|}$.
- Cas $[x]t$. Comme $([x]t)^\sigma = [x]t^\sigma$ et pour tout terme u $|[x]u| = |u|^\sigma$ nous avons $|([x]t)^\sigma| = |t^\sigma|$ et $(|t| + |\sigma|)^{|t|} = (|[x]t| + |\sigma|)^{|[x]t|}$. Or par hypothèse d'induction $|t^\sigma| \leq (|t| + |\sigma|)^{|t|}$, donc $|([x]t)^\sigma| = (|[x]t| + |\sigma|)^{|[x]t|}$.
- Cas $f(t_1, \dots, t_n)$. Nous avons $|f(t_1, \dots, t_n)^\sigma| = |f(t_1^\sigma, \dots, t_n^\sigma)| = 1 + \sum_i |t_i^\sigma|$. Par hypothèse d'induction, nous avons pour tout i $|t_i^\sigma| \leq (|t_i| + |\sigma|)^{|t_i|}$. Notons $k = \sum_i |t_i| = |f(t_1, \dots, t_n)| - 1$. Pour tout i nous avons en particulier $|t_i| \leq k$, donc $1 + \sum_i |t_i^\sigma| \leq 1 + n(k + |\sigma|)^k$. De même $n \leq k$, donc $1 + \sum_i |t_i^\sigma| \leq 1 + k(k + |\sigma|)^k$. De plus $1 \leq |\sigma|(k + |\sigma|)^k$, donc

$$\begin{aligned} |f(t_1, \dots, t_n)^\sigma| &\leq 1 + k(k + |\sigma|)^k \\ &\leq |\sigma|(k + |\sigma|)^k + k(k + |\sigma|)^k \\ &\leq (k + |\sigma|)(k + |\sigma|)^k \\ &\leq (k + |\sigma|)^{k+1} \\ &\leq (|f(t_1, \dots, t_n)| + |\sigma|)^{|f(t_1, \dots, t_n)|} \end{aligned}$$

- Cas $Z(t_1, \dots, t_n)$. Notons $\sigma(Z) = \lambda x_1 \dots x_n. u$. Nous avons $(Z(t_1, \dots, t_n))^\sigma = u^{\{x_1 := t_1^\sigma, \dots, x_n := t_n^\sigma\}}$. Alors par lemme 4.59 de taille d'une substitution nous avons

$$\begin{aligned} |u^{\{x_1 := t_1^\sigma, \dots, x_n := t_n^\sigma\}}| &= |u| + \sum_i |\text{fo}_{x_i}(u)|(|t_i^\sigma| - 1) \\ &= |u| - \sum_i |\text{fo}_{x_i}(u)| + \sum_i |\text{fo}_{x_i}(u)||t_i^\sigma| \end{aligned}$$

Par hypothèse d'induction nous avons pour i $|t_i^\sigma| \leq (|t_i| + |\sigma|)^{|t_i|}$. Notons $k = \sum_i |t_i| = |Z(t_1, \dots, t_n)| - 1$. Pour tout i nous avons en particulier $|t_i| \leq k$, donc $|t_i^\sigma| \leq (k + |\sigma|)^k$. De plus $\sum_i |\text{fo}_{x_i}(u)| \leq |u|$, donc $|u| - \sum_i |\text{fo}_{x_i}(u)| \geq 0$ et

$$\begin{aligned} |u^{\{x_1 := t_1^\sigma, \dots, x_n := t_n^\sigma\}}| &= (|u| - \sum_i |\text{fo}_{x_i}(u)|) + \sum_i |\text{fo}_{x_i}(u)||t_i^\sigma| \\ &\leq (|u| - \sum_i |\text{fo}_{x_i}(u)|) + \sum_i |\text{fo}_{x_i}(u)|(k + |\sigma|)^k \\ &\leq (|u| - \sum_i |\text{fo}_{x_i}(u)|)(k + |\sigma|)^k + \sum_i |\text{fo}_{x_i}(u)|(k + |\sigma|)^k \\ &\leq |u|(k + |\sigma|)^k - \sum_i |\text{fo}_{x_i}(u)|(k + |\sigma|)^k + \sum_i |\text{fo}_{x_i}(u)|(k + |\sigma|)^k \\ &\leq |u|(k + |\sigma|)^k \end{aligned}$$

Enfin, par définition $|u| \leq |\sigma| \leq k + |\sigma|$, donc

$$\begin{aligned} (Z(t_1, \dots, t_n))^\sigma &\leq |u|(k + |\sigma|)^k \\ &\leq (k + |\sigma|)^{k+1} \\ &\leq (|Z(t_1, \dots, t_n)| + |\sigma|)^{|Z(t_1, \dots, t_n)|} \end{aligned}$$

□

L'exposant dans la borne $(|r| + |\sigma|)^{|r|}$ correspond à la possibilité d'avoir dans r un emboîtement des méta-variables. L'exemple 4.61 démontre cette possibilité.

Exemple 4.61.

Considérons cette règle itérant quatre fois une fonction :

$$q : @(\langle x \rangle Z(x), a) \rightarrow Z(Z(Z(Z(a))))$$

Considérons la valuation partielle

$$\sigma = \{ Z \mapsto \underline{\lambda}x_1. @(x_1, x_1) \}$$

qui nous donne une règle du raffinement canonique de q

$$@(\langle x \rangle @(x, x), a) \rightarrow @(@(@(@(a, a), @(a, a)), @(@(a, a), @(a, a))), @(@(@(a, a), @(a, a)), @(@(a, a), @(a, a))))$$

dans laquelle l'explosion de taille liée à l'exposant est visible.

Avec le lemme 4.60 de taille d'une valuation nous savons donc que la taille du membre droit r^σ d'une règle du raffinement canonique d'une règle $l \rightarrow r$ est bornée par $(|r| + |\sigma|)^{|r|}$. Le paramètre $|r|$ est une constante pour toutes les règles du raffinement canonique de $l \rightarrow r$, et est donc borné dans tout CRS comportant un nombre fini de règles.

Le paramètre $|\sigma|$ en revanche dépend du raffinement de $l \rightarrow r$ choisi, et donc de la source $c[l^\sigma]$ du pas de réduction concerné. Par définition du raffinement canonique tout élément du codomaine de σ est de la forme $\underline{\lambda}x_1 \dots x_n. s[Z_1^s, \dots, Z_k^s]$ où s est inclus dans un squelette de l^σ . En particulier $|s[Z_1^s, \dots, Z_k^s]| = |s| + k$ et la taille $|\sigma|$ de la valuation partielle est bornée par les tailles des squelettes de l^σ augmentées de leurs nombres de trous.

Finalement, nous pourrions borner la croissance des graphes si nous sommes capables de borner la croissance des squelettes. Cependant, la taille des squelettes apparaissant dans une séquence de réduction d'un CRS faible n'est en général pas bornée polynomialement par la longueur de la séquence. L'exemple 4.63 illustre trois mécanismes pouvant faire exploser la taille des squelettes.

Convention 4.62. Nous noterons $f^n(t_1, \dots, t_k)$ l'application successive de n occurrences de f aux termes t_1, \dots, t_k définie par :

$$\begin{aligned} f^1(t_1, \dots, t_k) &= f(t_1, \dots, t_k) \\ f^{n+1}(t_1, \dots, t_k) &= f(f^n(t_1, \dots, t_k), \dots, f^n(t_1, \dots, t_k)) \end{aligned}$$

Exemple 4.63.

Considérons la règle de composition suivante :

$$c(\langle x \rangle Z(x)) \rightarrow \langle x \rangle Z(Z(x))$$

et le terme

$$c^n(\langle x \rangle g(x))$$

Alors nous avons la séquence de réduction de longueur n

$$c^n(\langle x \rangle g(x)) \rightarrow c^{n-1}(\langle x \rangle g^2(x)) \rightarrow c^{n-2}(\langle x \rangle g^4(x)) \rightarrow \dots \rightarrow \langle x \rangle g^{2^n}(x)$$

au terme de laquelle un squelette $\langle x \rangle g^{2^n}(x)$ de taille supérieure à 2^n est formé.

De même avec la règle :

$$a(\langle x \rangle Z(x)) \rightarrow \langle x \rangle Z(f(x, x))$$

et la séquence

$$a^n(\langle x \rangle x) \rightarrow a^{n-1}(\langle x \rangle f(x, x)) \rightarrow a^{n-2}(\langle x \rangle f(f(x, x), f(x, x))) \rightarrow \dots \rightarrow \langle x \rangle f^n(x, x)$$

et la règle

$$b(\langle x \rangle Z(x)) \rightarrow \langle x \rangle f(Z(x), Z(x))$$

et la séquence

$$b^n(\langle x \rangle x) \rightarrow b^{n-1}(\langle x \rangle f(x, x)) \rightarrow b^{n-2}(\langle x \rangle f(f(x, x), f(x, x))) \rightarrow \dots \rightarrow \langle x \rangle f^n(x, x)$$

Nous allons donc pour la suite nous restreindre à un ensemble de CRS dits *plats* dans lesquels la stabilité de la taille des squelettes est imposée. Pour énoncer cette restriction, nous étendons la notion de squelette aux méta-termes en considérant les méta-variables comme des symboles ordinaires.

Définition 4.64 (CRS plat). *Qualifions de **plat** un CRS dans lequel pour toute règle $\kappa : l \rightarrow r$, tout squelette de r apparaît à l'identique dans l .*

Exemple 4.65.

La règle de l'exemple 4.63

$$c(\langle x \rangle Z(x)) \rightarrow \langle x \rangle Z(Z(x))$$

n'est pas plate, car son membre droit contient un squelette $\langle x \rangle Z(Z(x))$ qui n'existe pas dans le membre gauche. De même pour la règle de composition

$$c(\langle x \rangle Z(x), \langle x \rangle Z'(x)) \rightarrow \langle x \rangle Z(Z'(x))$$

En revanche, la règle de l'exemple 4.61

$$@(\langle x \rangle Z(x), a) \rightarrow Z(Z(Z(a)))$$

est plate malgré son emboîtement de méta-variables, de même que la règle β habituelle et que les règles de filtrage par des motifs algébriques [KvOdV08]. Enfin, les exemples usuels que sont la règle de point fixe

$$\mu(\langle x \rangle Z(x)) \rightarrow Z(\mu(\langle x \rangle Z(x)))$$

et la règle d'itération correspondant à la fonction *fold*

$$F(\langle x, y \rangle Z(x, y), Z_0, Z_1 : Z_2) \rightarrow F(\langle x, y \rangle Z(x, y), Z(Z_0, Z_1), Z_2)$$

sont également plates, puisque les squelettes $\langle x \rangle Z(x)$ et $\langle x, y \rangle Z(x, y)$ de leurs membres droits apparaissent bien dans leurs membres gauches.

Nous pouvons maintenant combiner cette restriction de platitude avec le lemme 4.60 de taille des valuations pour borner la taille graphique de la cible d'une étape de réduction parallèle étiquetée en fonction de sa source.

Définition 4.66 (Taille des squelettes). Notons $|t|_s$ le nombre de symboles du plus grand squelette de t , trous compris. Formellement, soit S l'ensemble des squelettes des sous-termes de t , nous définissons alors $|t|_s = \max_{s \in S} (|s|)$ (où les trous sont comptés).

Lemme 4.67 (Croissance des graphes). Pour tout CRS faible plat Σ il existe une constante k_Σ telle que pour tout graphe \mathcal{G}_t , si $\mathcal{G}_t \Rightarrow \mathcal{G}_{t'}$ alors $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + (k_\Sigma + |t|_s)^{k_\Sigma}$.

Démonstration. Notons k_Σ la plus grande taille d'un membre droit de règle de Σ . Notons $l \rightarrow r$ la règle utilisée par l'étape de réduction parallèle étiquetée considérée. Les seules nouvelles étiquettes dans la cible sont apportées par r , et par définition le même r est utilisé pour tous les pas de réduction effectués en parallèle. Nous avons donc $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + |r|$.

En outre, par définition de l'étiquetage d'un CRS il existe une règle $l_0 \rightarrow r_0$ de Σ et une valuation partielle σ_0 appartenant au raffinement canonique de l telle que $l \rightarrow r$ est un étiquetage de la règle $l_0^{\sigma_0} \rightarrow r_0^{\sigma_0}$. Nous avons donc $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + |r_0^{\sigma_0}|$, et par lemme 4.60 $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + (|r_0| + |\sigma_0|)^{|r_0|}$. Or par définition de k_Σ nous avons $|r_0| \leq k_\Sigma$, et par définition d'un raffinement canonique $|\sigma_0| \leq |l_0^\sigma|_s \leq |t|_s$. Donc $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + (k_\Sigma + |t|_s)^{k_\Sigma}$. \square

Lemme 4.68 (Croissance le long d'une séquence). Pour tout CRS faible plat Σ il existe une constante k_Σ telle que pour tout graphe \mathcal{G}_t et toute séquence de réduction de graphes $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_{t'}$ de n étapes nous avons $|t'|^\mathcal{G} \leq |t|^\mathcal{G} + n(k_\Sigma + |t|_s)^{k_\Sigma}$.

Démonstration. Par induction sur n avec le lemme 4.67, et car la taille des squelette est invariante par définition d'un CRS plat. \square

En partant d'un terme étiqueté initial, qui vérifie nécessairement la propriété de partage et représente donc un graphe, et en remarquant que la taille d'un squelette est bornée par la taille du terme, nous obtenons le corollaire :

Corollaire 4.69. *Pour tout CRS faible plat Σ il existe une constante k_Σ telle que pour tout terme étiqueté initial t et toute séquence de réduction de graphes $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_t'$ de n étapes nous avons $|t'|^\mathcal{G} \leq |t| + n(k_\Sigma + |t|)^{k_\Sigma}$.*

Nous en déduisons immédiatement le théorème.

Théorème 4.70 (Complexité dérivationnelle). *Si t est un terme étiqueté initial, alors le coût Turing d'une séquence de réduction de graphe $\rho : \mathcal{G}_t \Rightarrow \mathcal{G}_t'$ dans un CRS faible plat est polynomial en $|t|$ et la longueur de ρ .*

Ainsi, pour les CRS plats notre borne polynomiale sur le coût réel de l'évaluation faible partagée pleinement paresseuse n'est plus fonction que de la longueur des séquences de réduction parallèle étiquetée et de la taille du terme de départ. Ceci donne une première validation de l'adéquation de notre modèle abstrait (la réduction pleinement paresseuse dans les CRS) avec le réel, qui indique la portée des investigations à venir dans les prochains chapitres.

Notons toutefois que ce lien est largement perfectible. Un lien polynomial avec une machine de Turing assure à notre modèle abstrait des racines dans le réel qui restent superficielles, car la machine de Turing est moins proche d'un ordinateur réel que ne le sont la plupart des machines virtuelle à registres ou même à pile, et car un lien polynomial laisse encore beaucoup de place à l'incertitude.

Bilan du chapitre

Dans ce chapitre nous avons proposé une généralisation de la notion de partage pleinement paresseux au cadre des CRS. Cette généralisation est décrite par l'intermédiaire d'une méthode d'étiquetage pour un CRS arbitraire (fût-il non-orthogonal), et nous avons prouvé que cet étiquetage généralisé définissait sur le λ -calcul la même réduction partagée que la technique de Wadsworth [Wad71] (théorème 4.52 de partage équivalent).

Nous avons profité de la nature extensible du cadre des CRS pour faire porter les mécanismes des étiquettes non par la méta-théorie, mais par une transformation de la signature et de l'ensemble de règle d'un CRS. Ainsi, un CRS étiqueté n'est qu'un CRS comme les autres. Pour ce faire nous avons introduit une notion de raffinement d'un CRS et avons isolé en particulier un raffinement dit canonique qui est la manière la plus économe de transformer tout CRS en un autre CRS équivalent dont les règles n'utilisent que des méta-variables d'arité 0 (section 4.2). Nous avons ensuite pu proposer une technique d'étiquetage qui construit un SLS à partir d'un CRS raffiné.

Enfin, nous avons proposé une mesure de coût pour la réduction partagée pleinement paresseuse valable dans l'ensemble des CRS (définition 4.54) et isolé un fragment des CRS (les CRS plat, définition 4.64) dans lequel le simple nombre d'étapes de réduction parallèle étiquetée a un rapport polynomial avec le coût d'exécution sur une machine de Turing (théorème 4.70 de complexité dérivationnelle).

TROISIÈME PARTIE

CAUSALITÉ

Introduction

La présente partie inspecte les relations entre la notion unifiée de partage pleinement paresseux qui a émergé de la partie précédente et la notion de réduction faible (dont il convient de rappeler qu'elle est la réduction faible combinatoire et non la réduction faible naïve, par convention 0.53¹).

L'enseignement principal de cette partie est le suivant : le partage pleinement paresseux partage exactement les pas de réduction qui sont considérés comme « le même événement » dans une théorie des résidus à la Lévy appliquée au cadre de la réduction faible. Cette conclusion est un indice important dans la caractérisation de la pleine paresse comme la réduction optimale dans un espace de réduction faible, et une étape clé dans la démonstration de ce fait (qui sera achevée au chapitre 10).

Le chapitre 5 introduit l'interprétation de la réécriture en termes d'événements et de causalité dans les cas simples du λ -calcul et du λ -calcul faible et montre comment en déduire des étiquetages remplissant la double fonction de :

- témoigner de la structure causale du calcul, et
- si possible, fournir un *SLS* dont la réduction parallèle partage les pas de réduction causalement équivalents.

Il est remarqué en particulier que l'étiquetage causal obtenu pour le λ -calcul faible est équivalent à l'étiquetage définissant le partage pleinement paresseux.

Le chapitre 6 sort du cas rassurant et orthogonal du λ -calcul pour porter les discussions de causalité dans le système plus riche qu'est le *Pure Pattern Calculus*, un calcul qui généralise le λ -calcul en apportant des mécanismes de filtrage. Ce cadre permet d'observer comment le défaut d'orthogonalité en général et le défaut de stabilité en particulier rendent plus complexe la structure causale d'un système de réécriture.

Le chapitre 7 enfin poursuit l'analyse de la causalité dans le cadre général des *CRS*, et sépare les propriétés causales vues au chapitre 5 avec le λ -calcul selon qu'elles peuvent être transmises à tout *CRS* ou uniquement aux *CRS* orthogonaux.

La conclusion de cette longue analyse nous montre que, si le partage pleinement paresseux peut aussi bien être défini dans des cadres orthogonaux et non-orthogonaux, il ne peut en revanche être mis en relation avec une notion d'optimalité que dans le cadre orthogonal. En dehors de ce cadre en effet une bonne théorie des réductions optimales manque encore.

1. faute de quoi les observations faites dans cette partie concerneraient le partage paresseux et non le partage pleinement paresseux

CHAPITRE 5

Le cas du λ -calcul

Sommaire

5.1 Causalité dans le λ-calcul	150
5.2 Causalité en réduction faible	153
5.3 Lambda-calcul faible étiqueté	155
5.4 Correction et complétude causales	160
5.5 Discussion : variantes de forme	163
5.6 Discussion : causalités	166
5.7 Discussion : reconstruction	170
5.8 Discussion : une autre preuve de partage	174

Les systèmes de partage étiquetés (*SLS*) définis à la partie I ont pour vocation principale la représentation de graphes acycliques par des termes étiquetés, et la représentation de séquences de réduction de graphes par des séquences de réduction de termes étiquetés. Cependant, certains des systèmes étiquetés apparaissant dans la littérature possèdent d'autres caractéristiques en plus de cette propriété de partage.

Dans le λ -calcul faible étiqueté de Blanc, Lévy et Maranget [BLM07] par exemple, si un pas de réduction ρ_1 crée un pas de réduction ρ_2 , alors les étiquettes du radical de ρ_2 sont liées aux étiquettes du radical de ρ_1 , ce qui nous donne quelques informations sur les relations de dépendance entre les différents pas de réduction. Le travail de Maranget sur les systèmes du premier ordre orthogonaux [Mar91, Mar92] pousse plus loin les résultats sur les étiquettes, puisqu'il démontre que son étiquetage permet de définir des réductions optimales au sens de Lévy.

Dans ce chapitre nous allons disséquer les relations de nécessité entre pas de réduction dans le λ -calcul faible, et poursuivre l'analyse de Blanc, Lévy et Maranget pour démontrer que leur λ -calcul faible étiqueté [BLM07] reflète précisément ces relations de nécessité.

La section 5.1 rappelle la classification des créations de radicaux en λ -calcul donnée par Lévy [Lé78], puis la section 5.2 étend cette classification au λ -calcul faible. La section 5.3 construit une légère variante du système de Blanc, Lévy et Maranget en se basant sur les enseignements de la section 5.2, et enfin la

section 5.4 démontre les propriétés causales de ces étiquettes. Les section 5.5, à 5.8 discutent ensuite les différences de forme entre l'étiquetage de ce chapitre et ceux des chapitres 2 et 3, les possibilités d'anticiper ou retarder la propagation d'informations causales, des propriétés des étiquettes plus fortes que la causalité, et les différences entre les preuves de préservation de la propriété de partage données par cette thèse et par Blanc, Lévy et Maranget [BLM07].

5.1 Causalité dans le λ -calcul

Lévy [Lé78] a caractérisé les réductions optimales du λ -calcul au moyen d'une notion de familles de radicaux formalisant une idée de « radicaux ayant une identité commune » ou « radicaux représentant le même événement ». Dans cette approche, un radical est caractérisé par l'ensemble des causes de son existence. Ainsi, deux radicaux r_1 et r_2 seront considérés de la même famille s'ils ont une origine commune et si les mêmes événements de réduction passés ont *contribué* à faire de r_1 et r_2 des radicaux.

La technique a ensuite deux piliers :

1. Identifier les contributions directes à travers l'étude des cas dans lesquels un pas de réduction peut créer un nouveau radical.
2. Enregistrer et composer les liens de contribution directe afin d'obtenir l'ensemble des contributions directes ou indirectes à la création de chaque radical. Des étiquettes seront utilisées pour ceci.

Commençons par l'étude des créations de radicaux dans le λ -calcul. La figure 5.1 montre des exemples de contributions dans ce cadre. Rappelons que les radicaux du λ -calcul sont exactement les sous-termes de la forme $(\lambda x.t)u$, indépendamment de leur environnement. Nous voulons donc savoir les manières dont de tels sous-termes peuvent être formés au sein d'un λ -terme. Lévy a proposé une classification en trois cas [Lé78] illustrés par les exemples suivants.

Exemple 5.1.

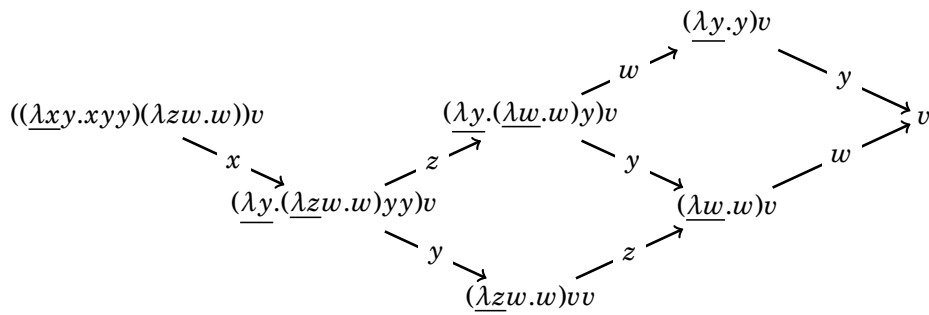
Les trois réductions suivantes montrent les trois manières de mettre en contact une λ -abstraction $\lambda x.t$ et un argument u décrites par Lévy.

$$\begin{aligned} ((\lambda y.\lambda x.t)v)u &\rightarrow (\lambda x.t^{x:=v})u \\ ((\lambda y.y)\lambda x.t)u &\rightarrow (\lambda x.t)u \\ (\lambda y.yu)\lambda x.t &\rightarrow (\lambda x.t)u \end{aligned}$$

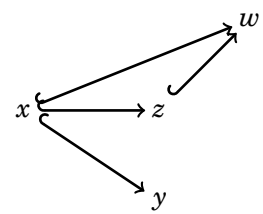
Remarquons que le premier cas est celui qui est en œuvre à la figure 5.1 pour la création d'un radical y par un radical x et d'un radical w par un radical z . Le troisième cas apparaît lui pour la création d'un radical z par un radical x .

Pour nos besoins présents, une vue un peu plus grossière est suffisante, et nous pouvons ramener la classification de Lévy à deux cas : en λ -calcul un radical est créé au moment où une abstraction λx arrive au contact d'une application. D'une part ceci peut venir d'une réduction du sous-terme à gauche de l'application du radical créé, qui amène l'abstraction "par en-dessous" (1).

Dans les termes de l'espace de réduction suivant les λ -abstractions définissant des radicaux sont soulignées.



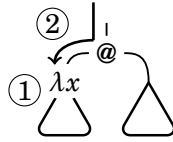
Dans cet espace de réduction, chaque λ -abstraction correspond à une famille de radicaux. Les dépendances causales entre ces quatre familles sont résumées dans le graphe de dépendance suivant, où la flèche \hookrightarrow dénote une contribution entre deux familles.



Remarquons que la famille x crée la famille z qui crée la famille w , d'où une dépendance indirecte entre x et w .

FIGURE 5.1: Contributions directes et indirectes

Ce cas couvre les deux premiers cas de Lévy. D'autre part, l'abstraction peut être apportée par une substitution extérieure, en remplacement d'une certaine occurrence de variable (2). Ceci est exactement le troisième cas de Lévy.



Exemple 5.2.

Le cas de création (1) couvre les deux premiers cas de Lévy :

$$\begin{aligned} ((\lambda y. \lambda x. t)v)\underline{u} &\rightarrow (\lambda x. t^{x:=v})\underline{u} \\ ((\lambda y. y)\underline{\lambda x. t})\underline{u} &\rightarrow (\lambda x. t)\underline{u} \end{aligned}$$

tandis que le cas (2) est le troisième cas de Lévy :

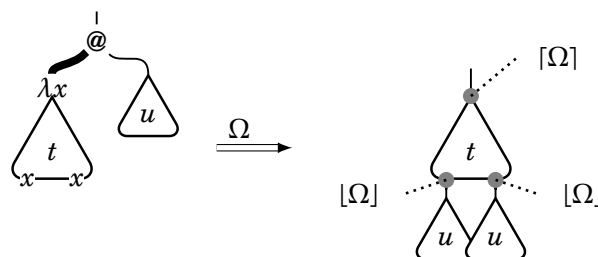
$$(\lambda y. y\underline{u})\underline{\lambda x. t} \rightarrow (\lambda x. t)\underline{u}$$

Un étiquetage du λ -calcul peut être construit en inversant la précédente description. Nous passons d'une approche tournée vers le passé où la question est *d'où ce radical vient-il ?* à une approche tournée vers le futur où la question devient *quelles peuvent être les conséquences de ce pas de réduction ?* Si chaque radical réduit laisse son empreinte à tout endroit où il peut contribuer à quelque chose (et à aucun autre endroit !), alors l'origine d'un radical peut être reconstituée par une collecte des informations qui ont été laissées à cette place par les pas de réduction passés.

Pour réaliser ceci, Lévy a ajouté des étiquettes à la syntaxe du λ -calcul, de manière que chaque étiquette enregistre les informations localement pertinentes à propos des réductions passées, et que les réductions agissent sur ces étiquettes pour maintenir les informations à jour.

Une différence par rapport aux étiquetages des chapitres 2 à 4 est qu'un *nom* (« degré » dans la version originale) est attribué à chaque pas de réduction. Ce nom est déduit des étiquettes du radical et sert à marquer les potentielles contributions futures.

La réduction d'un radical r de nom Ω étend les étiquettes du réduit de deux manières qui correspondent aux deux cas de création. D'abord r peut contribuer à quelque chose à sa racine, par le cas (1). Une étiquette $[\Omega]$ est donc incorporée à l'étiquette de la racine du réduit. De plus, r peut contribuer à quelque chose par le cas (2) aux positions des variables substituées. Une étiquette $[\Omega]$ est incorporée aux étiquettes des termes substitués pour rendre compte de ce phénomène. Le schéma informel suivant visualise cette discipline, et sera à comparer avec celui de la section suivante figurant la causalité en réduction faible.



Dans un souci d'homogénéité avec la suite, les notations $[\Omega]$ et $[\underline{\Omega}]$ remplacent respectivement dans cette explication les $\bar{\beta}$ et $\underline{\beta}$ utilisés originellement par Lévy.

5.2 Causalité en réduction faible

La restriction du λ -calcul à la réduction faible modifie sa structure causale. Notamment, un nouveau cas de création de radical intervient dans le calcul faible. Après analyse de cette différence, nous présentons une version à peine modifiée du λ -calcul faible étiqueté de Blanc, Lévy et Maranget [BLM07].

Supposons en effet qu'un certain sous-terme r a déjà la forme d'un radical (c'est donc un pré-radical), mais n'en est pas un dans le formalisme faible pour la seule raison qu'il contient une occurrence de variable liée à l'extérieur. Le pré-radical r peut alors devenir un radical au moment où cette occurrence est substituée. Après une première présentation informelle de ce nouveau phénomène le lemme 5.4 donnera un énoncé complet des possibilités de création de β -radicaux faibles.

Le nouveau cas est capturé par une généralisation du cas (2) que nous noterons (2⁺). Le cas (2) était celui d'une variable à gauche d'une application substituée par une λ -abstraction. Le cas (2⁺) est quant à lui celui d'une application affectée par une substitution quelconque, le seul critère étant que la substitution modifie effectivement le sous-terme (en d'autres termes, le sous-terme considéré doit contenir au moins une occurrence libre de la variable substituée).

Exemple 5.3.

Cas de la réduction faible : à la source du pas de réduction suivante, un pré-radical $(\lambda x.yx)\underline{u}$ n'est pas un radical dans le formalisme faible pour la seule raison de la présence d'une occurrence libre de la variable y liée au-dessus. Cependant, après le pas de réduction cette restriction disparaît et le pré-radical devient un radical.

$$(\lambda y.((\lambda x.yx)\underline{u}))t \rightarrow (\lambda x.tx)u$$

Ceci généralise (2) et contient encore le troisième cas de Lévy :

$$(\lambda y.y\underline{u})\lambda x.t \rightarrow (\lambda x.t)u$$

Le lemme suivant donne un énoncé précis de cette classification des créations, qui nous permettra de dériver formellement certaines propriétés causales du λ -calcul faible étiqueté présenté à la section suivante.

Lemme 5.4 (Création de radicaux). *Soit un pas de réduction $\rho : t = c[r] \rightarrow c[r'] = t'$ avec $r = (\lambda x.u)v$ et un radical $r_c = (\lambda y.u_c)v_c$ de t' créé par ρ . Notons $t' = c_c[r_c]$. Alors l'une des propositions suivantes est vérifiée.*

$$(1) \quad c = c_c[\square v_c] \text{ et } u^{\{x:=v\}} = \lambda y.u_c.$$

(2⁺) *La position de r_c descend d'une position de la $\{x\}$ -épine de u .*

Pour la preuve, commençons par isoler le cas de la création d'un radical à la racine du terme.

Lemme 5.5. *Sous les hypothèses du lemme 5.4, si de plus $c_c = \square$ et $c \neq \square$ alors la proposition (1) est vérifiée.*

Démonstration. Remarquons d'abord que $c_c = \square$ implique $r_c = t' = c[u^{\{x:=v\}}]$. Raisonnons par cas sur $c \neq \square$:

- Cas $c = \lambda z.c'$. Alors il existe u' tel que $t' = \lambda z.u'$. Ceci n'est pas notre cas car $\lambda z.u'$ n'est pas un pré-radical.
- Cas $c = (\lambda y.u_c)c'$. Alors l'ancêtre t de r_c est un radical. Ceci n'est pas notre cas.
- Cas $c = c'v_c$. Raisonnons par cas sur c' .
 - Cas $c' = \square$, c'est-à-dire $c = \square v_c$. Alors $u^{\{x:=v\}} = \lambda y.u_c$ et la proposition (1) est vérifiée.
 - Cas $c' = t_1c''$ ou $c' = c''t_2$, c'est-à-dire $c = t_1c''v_c$ ou $c = c''t_2v_c$. Alors il existe une paire t'_1, t'_2 telle que $t' = t'_1t'_2v_c$. Ceci n'est pas notre cas.
 - Cas $c' = \lambda y.c''$, c'est-à-dire $c = (\lambda y.c'')v_c$. Alors l'ancêtre t de r_c est déjà un radical. Ceci n'est pas le cas. \square

Démonstration du lemme 5.4. Raisonnons par cas sur les positions relatives de r' et r_c .

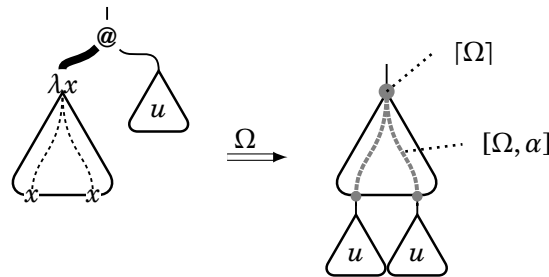
- Si r' et r_c sont disjoints, alors le radical r_c existe déjà dans t . Ceci n'est pas notre cas.
- Si r' est un sous-terme de r_c , alors il existe c' tel que $r_c = c'[r']$ et $t' = c_c[c'[r']]$. Comme r_c est créé par ρ , son ancêtre $c'[r]$ dans n'est pas un radical dans $t = c_c[c'[r]]$. Supposons que $c'[r]$ soit un pré-radical gelé. Alors il existe dans $c'[r]$ une occurrence d'une variable x liée par un lieu de c_c . Raisonnons par cas sur la position de cette variable.
 - Si x apparaît dans r , alors le pré-radical r est lui-même gelé, ce qui n'est pas notre cas.
 - Si x apparaît dans c' , alors le pré-radical $c'[r']$ contient toujours x et est donc toujours gelé, ce qui n'est pas notre cas.

Donc $c'[r]$ n'est pas un pré-radical gelé, et nous pouvons appliquer le lemme 5.5 au pas de réduction $c'[r] \rightarrow c'[r']$.

- Si r_c est un sous-terme de $r' = u^{\{x:=v\}}$, alors trois cas sont à considérer.

- Si r_c est dans une occurrence substituée de v , alors son ancêtre est déjà un radical. Ceci n'est pas notre cas.
- Si la position de r_c est dans la $\{x\}$ -épine de u , alors la proposition (2^+) est vérifiée.
- Si la position de r_c est dans u mais n'est pas dans la $\{x\}$ -épine de u , alors l'ancêtre de r_c lui est égal et est déjà un radical. Ceci n'est pas notre cas. \square

La réduction d'un radical r de nom Ω transforme les étiquettes du réduit. Comme précédemment, r peut contribuer à quelque chose à sa racine, par le cas (1). Une étiquette $[\Omega]$ dénotant l'épicentre de Ω est ajoutée à la racine du réduit pour témoigner de ceci. De plus, r peut contribuer à quelque chose par le cas (2^+) , à n'importe quelle position le long du chemin de propagation de la substitution, qui correspond à une épine dans le vocabulaire des chapitres précédents. Cet autre type de contribution sera marqué dans les étiquettes par la construction $[\Omega, \alpha]$ que nous avons déjà vue, qui dénote rappelons-le une copie de α déclenchée par Ω . Le schéma suivant résume ceci :



Remarquons au passage que les étiquettes $[\Omega]$ évoquées à la section 5.1 pour tenir compte du cas de création (2) n'apparaissent plus ici. Elles sont en effet couvertes par les étiquettes $[\Omega, \alpha]$ liées au cas (2^+) qui généralise le cas (2).

5.3 Lambda-calcul faible étiqueté

Nous allons maintenant formaliser le λ -calcul faible étiqueté décrit ci-dessus, ainsi que les propriétés causales de ses étiquettes. Nous verrons aussi que ledit calcul étiqueté est un *SLS*.

5.3.1 Définition

La première différence entre le λ -calcul étiqueté présenté dans cette section et ceux des chapitres 2 et 3 est la position des étiquettes. Alors que les étiquettes étaient dans les chapitres précédents portées par les symboles de la syntaxe d'origine, elles seront ici des symboles à part entière et à ce titre auront notamment leurs propres positions.

Dans la grammaire ci-dessous, la catégorie des termes étiquetés dénote de manière générale les λ -termes augmentés des étiquettes évoquées à la section précédente. Deux catégories syntaxiques à part, les termes étiquetés complets

ou incomplets, caractérisent respectivement les termes étiquetés qui ont ou n'ont pas une étiquette à leur racine.

Définition 5.6 (Étiquettes et termes étiquetés). *Les étiquettes et les termes étiquetés sont définis par la grammaire suivante.*

$$\begin{array}{lll}
\alpha, \beta, \dots, \omega & ::= & p \mid [\Omega] \mid [\Omega, \alpha] & \text{Étiquettes} \\
\Gamma, \Delta, \Omega & ::= & \alpha_1 \alpha_2 \dots \alpha_n & \text{Séquences d'étiquettes} \\
X, Y, Z & ::= & T \mid N & \text{Termes étiquetés} \\
N & ::= & TT \mid \lambda x. T & \text{Termes étiquetés incomplets} \\
T & ::= & x \mid \alpha : X & \text{Termes étiquetés complets}
\end{array}$$

où n est un entier strictement positif, $x, y, z \in \mathcal{X}$ sont des variables, et p est une position, qui comme dans les chapitres précédents sert d'étiquette initiale. Comme précédemment, un **terme initial** est un terme dont les étiquettes sont toutes initiales et différentes. Pour toute séquence d'étiquettes $\Gamma = \gamma_1 \dots \gamma_n$ et tout terme X , nous abrégons $\gamma_1 : \dots : \gamma_n : X$ en $\Gamma \cdot X$. Nous notons encore \mathcal{L} ce nouvel ensemble d'étiquettes.

Dans cette grammaire les variables sont considérées comme des termes étiquetés complets, et ne sont donc pas forcées de porter une étiquette. Nous retrouvons donc sur ce point les choix qui ont été faits aux chapitres précédents. Notons cependant que les variables peuvent être précédées d'étiquettes car tous les termes étiquetés, complets ou non, le peuvent. La section 5.7 discutera quelques choix possibles sur ce point.

La lettre grecque Ω indique une séquence d'étiquettes jouant le rôle de nom d'un radical. Les étiquettes peuvent ici être vues comme des membres à part entière de la signature du système, et la notion de position sur des termes syntaxiques les prend donc en compte.

Exemple 5.7.

Considérons l'étiquetage suivant d'un terme $\lambda y. ty$:

$$[\zeta \alpha] : [\zeta \alpha, \beta] : \lambda y. [\zeta \alpha, \gamma] : @([\zeta \alpha, \eta] : T, \kappa : y)$$

où T est un étiquetage quelconque du terme t . L'ensemble de ses positions est

$$\{\epsilon, 1, 11, 111, 1111, 1112, 11121\} \cup 11111 \cdot \text{pos}(T)$$

Par exemple, le symbole à la position 1 est $[\zeta \alpha, \beta]$ et celui à la position 11 est λy . Le sous-terme à la position 1112 est $\kappa : y$.

Les notions de variables libres ou liées, de substitution, d'épine, et de réduction faible sont héritées du λ -calcul.

La relation de *contribution directe* \hookrightarrow^d est une relation sur les séquences d'étiquettes. Elle sera utile dans le développement à venir notamment quand elle concernera des séquences d'étiquettes qui sont des noms de radicaux. La notion apparaît d'abord dans la thèse de Luc Maranget [Mar92] dans le cadre de systèmes de réécriture de termes du premier ordre, et adaptée à notre cadre de la façon suivante.

Définition 5.8 (Contribution). Une séquence d'étiquettes Ω **contribue directement** à une séquence d'étiquettes Γ , noté $\Omega \hookrightarrow^d \Gamma$, si et seulement s'il existe deux séquences d'étiquettes éventuellement vides Γ_1 et Γ_2 telles que $\Gamma = \Gamma_1[\Omega]\Gamma_2$ ou $\Gamma = \Gamma_1[\Omega, \alpha]\Gamma_2$. Nous notons \hookrightarrow et appelons relation de **contribution** la clôture transitive de \hookrightarrow^d .

Exemple 5.9.

Voici deux exemples de contribution directe :

$$\begin{array}{l} \Omega \hookrightarrow^d \alpha[\Omega] \\ \alpha[\Omega] \hookrightarrow^d [\alpha[\Omega], \beta]\gamma \end{array}$$

dont nous pouvons déduire la contribution indirecte

$$\Omega \hookrightarrow [\alpha[\Omega], \beta]\gamma$$

Dans le λ -calcul étiqueté de Blanc, Lévy et Maranget [BLM07], le nom d'un pas de réduction est propagé le long de l'épine de la λ -abstraction principale de son radical, ainsi qu'il était fait avec le β -système parcimonieux \mathcal{P}_{BLM} au chapitre 3.

Comme ici les étiquettes augmentent la structure des termes, précisons la définition des épines et des squelettes sur les termes du λ -calcul faible étiqueté :

Définition 5.10 (Épines étiquetées). La x -**épine** d'un λ -terme étiqueté X , notée $\langle X \rangle^x$, est le préfixe de X défini ainsi :

$$\begin{array}{l} \langle X \rangle^x = \square \quad x \notin \text{fv}(X) \\ \text{Sinon :} \\ \langle \alpha : X \rangle^x = \alpha : \langle X \rangle^x \\ \langle x \rangle^x = x \\ \langle \lambda y . T \rangle^x = \lambda y . \langle T \rangle^x \quad y \neq x \\ \langle T_1 T_2 \rangle^x = \langle T_1 \rangle^x \langle T_2 \rangle^x \end{array}$$

L'**épine** d'une λ -abstraction $\lambda x . T$ est le préfixe $\lambda x . \langle T \rangle^x$.

Définition 5.11 (Squelettes étiquetés). Le θ -**squelette** d'un λ -terme étiqueté X , noté $\langle\langle X \rangle\rangle^\theta$, est le préfixe de X défini ainsi :

$$\begin{array}{l} \langle\langle X \rangle\rangle^\theta = \square \quad \theta \cap \text{fv}(X) = \emptyset \\ \text{Sinon :} \\ \langle\langle \alpha : X \rangle\rangle^\theta = \alpha : \langle\langle X \rangle\rangle^\theta \\ \langle\langle x \rangle\rangle^\theta = x \quad x \in \theta \\ \langle\langle \lambda x . T \rangle\rangle^\theta = \lambda x . \langle\langle T \rangle\rangle^{\theta \cup \{x\}} \\ \langle\langle T_1 T_2 \rangle\rangle^\theta = \langle\langle T_1 \rangle\rangle^\theta \langle\langle T_2 \rangle\rangle^\theta \end{array}$$

Le **squelette** d'une λ -abstraction $\lambda x . T$ est le préfixe $\lambda x . \langle\langle T \rangle\rangle^{\{x\}}$.

De même, le réétiquetage uniforme $[\Omega, X]$ d'un contexte d'arité arbitraire X se transpose au cadre du λ -calcul faible étiqueté de ce chapitre :

$$\begin{aligned} [\Omega, \alpha : X] &= [\Omega, \alpha] : [\Omega, X] \\ [\Omega, \square] &= \square \\ [\Omega, x] &= x \\ [\Omega, \lambda x.T] &= \lambda x.[\Omega, T] \\ [\Omega, T_1 T_2] &= [\Omega, T_1][\Omega, T_2] \end{aligned}$$

Définition 5.12 (Règles de réduction étiquetées). *Un **pré-radical étiqueté** est un terme étiqueté complet de la forme*

$$R = \alpha : ((\Gamma \cdot \lambda x.T)U)$$

*Un pré-radical étiqueté R dans un terme X est un **radical étiqueté** quand sa position n'est pas gelée. Dans ce cas nous attribuons le nom $\Omega = \alpha\Gamma$ à tout pas de réduction dont le radical est R . Notons $T = S[T_1, \dots, T_n]$ avec $S = \langle T \rangle^x$.*

$$\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U) \rightarrow [\Omega] : [\Omega, S]^{x:=U}[T_1, \dots, T_n]$$

Exemple 5.13.

Le terme suivant est un radical étiqueté

$$\zeta : @(\alpha : \lambda x.\beta : \lambda y.\gamma : @(\eta : x, \kappa : y), T)$$

de nom $\zeta\alpha$ qui se réduit en

$$[\zeta\alpha] : [\zeta\alpha, \beta] : \lambda y.[\zeta\alpha, \gamma] : @([\zeta\alpha, \eta] : T, \kappa : y)$$

Notons que l'attribution d'étiquettes aux variables n'a rien d'impératif, et a pour seul but dans cet exemple de souligner le fait que la modification des étiquettes se fait le long d'une épine.

5.3.2 Propriété de partage

Blanc, Lévy et Maranget ont prouvé que le λ -calcul faible étiqueté précédent peut être doté d'une notion de réduction parallèle qui préserve une propriété de partage [BLM07]. Nous vérifions ici que ce résultat peut être obtenu à nouveau par vérification des axiomes des *SLS*. Cette vérification reprend essentiellement celle faite au chapitre 3 pour les β -systèmes parcimonieux.

Pour cela nous commençons par adapter l'ordre sur les étiquettes proposé à la définition 2.15 à la présence de séquences d'étiquettes.

Définition 5.14 (Ordre sur les étiquettes). *L'ensemble \mathcal{L} des étiquettes du λ -calcul faible étiqueté est muni de l'ordre partiel \leq généré par :*

$$\begin{aligned} \omega_1 &\leq [\omega_1 \dots \omega_n] \\ \omega_1 &\leq [\omega_1 \dots \omega_n, \alpha] \end{aligned}$$

Cet ordre ne prend en compte que la première étiquette d'une séquence. Remarquons que si la séquence d'étiquettes est le nom d'un pas de réduction, alors cette première étiquette qui seule compte est l'étiquette que porte la racine du radical du pas de réduction, c'est-à-dire l'étiquette sur laquelle porte l'axiome *Progression*.

Comme au chapitre 2, le respect local de la convention de Barendregt (c'est-à-dire le respect de la convention de Barendregt à l'intérieur de chaque squelette uniquement, définition 2.12) est préservé par la réduction faible dans ce λ -calcul étiqueté :

Lemme 5.15 (Convention de Barendregt locale). *Soit $X \rightarrow X'$ une étape de β -réduction faible étiquetée. Si $\mathbb{B}_l(X)$ alors $\mathbb{B}_l(X')$.*

Dans la définition des *SLS* au chapitre 1 nous utilisons une signature étiquetée dont les symboles sont formés par conjonction d'un symbole non-étiqueté et d'une étiquette. Pour faire entrer dans ce cadre les étiquettes de ce chapitre qui sont considérées comme des symboles à part entière nous utilisons la convention suivante.

Convention 5.16. *Nous utilisons donc un symbole factice ι d'arité 1 et considérons dans cette section l'expression $\alpha : X$ comme une notation pour l'expression $\iota^\alpha(X)$.*

Instanciation 5.17. *Considérons le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- la signature $\mathcal{S} = \mathcal{X} \cup \{\iota^\alpha \mid \alpha \in \mathcal{L}\} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{\@\}$, où les étiquettes comme les λ -abstractions sont considérées comme des symboles d'arité 1.
- l'ensemble \mathcal{T} des termes étiquetés respectant localement la convention de Barendregt.
- l'ensemble \mathcal{R} des pas de réduction faible étiquetée dont la source respecte localement la convention de Barendregt.

Pour tout pas de réduction $\rho \in \mathcal{R}$, notons

$$\rho : c[\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U)] \rightarrow c[[\alpha\Gamma] : [\alpha\Gamma, S]^{x:=U}[T_1, \dots, T_n]]$$

et définissons de même que précédemment :

- $\text{dex}(\rho) = \alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U)$
- $\text{duct}(\rho) = [\alpha\Gamma] : [\alpha\Gamma, S]^{x:=U}[T_1, \dots, T_n]$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \{e\} \cup 1 \cdot (\text{pos}(S) \setminus \text{fo}_x(S))$

Lemme 5.18. *Le système $\Sigma = (\mathcal{S}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration.

Source & Cible Par lemme 5.15.

Résidu L'ancêtre d'une position gelée est lui-même gelé, donc tout descendant d'un radical (non gelé) est toujours un radical.

Zone d'effet Soit un pas de réduction

$$\rho : c[\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U)] \rightarrow c[[\alpha\Gamma] : [\alpha\Gamma, S]^{x:=U}][T_1, \dots, T_n]$$

Par définition de $\text{effz}(\rho)$, toute position hors de la zone d'effet désigne soit un sous-terme d'un T_i soit un sous-terme de U . \square

Lemme 5.19. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine La racine de tout radical est une étiquette.

Progression Toute étiquette de la zone d'effet est de la forme $[\Omega]$ ou $[\Omega, \alpha]$ où Ω est le nom du radical. Par définition, Ω est une séquence d'étiquettes commençant par $\tau(\rho)$. Donc $\tau(\rho) \leq [\Omega]$ et $\tau(\rho) \leq [\Omega, \alpha]$.

Héritage Soient un pas de réduction $\rho \in \mathcal{R}$ et deux étiquettes $\alpha, \beta \in \mathcal{L}$ tels que β est une étiquette de la zone d'effet de ρ , $\alpha \neq \tau(\rho)$, et $\alpha < \beta$. En particulier β est de la forme $[\omega_1 \dots \omega_n]$ ou $[\omega_1 \dots \omega_n, \gamma]$ avec $\omega_1 = \tau(\rho)$. Donc $\alpha \leq \omega_1$ et même $\alpha < \omega_1$.

Partage Soit un pas de réduction

$$\rho : c[\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U)] \rightarrow c[[\alpha\Gamma] : [\alpha\Gamma, S]^{x:=U}][T_1, \dots, T_n]$$

Supposons $\mathbb{S}(\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U))$. En omettant sa racine qui est une étiquette $[\Omega]$, la zone d'effet de la réduction a la forme $[\Omega, S]$ où S est la x -épine de $S[T_1, \dots, T_n]$ et en particulier S est inclus dans un squelette de la source.

Soient deux positions $p, q \in \text{effz}(\rho) \setminus \{e\}$. En particulier $p, q \in \text{pos}(S)$, $\tau_p([\Omega, S]) = [\Omega, \tau_p(S)]$, et $\tau_q([\Omega, S]) = [\Omega, \tau_q(S)]$. Supposons $\tau_p([\Omega, S]) = \tau_q([\Omega, S])$. Alors $\tau_p(S) = \tau_q(S)$. Or $\mathbb{S}(\alpha : ((\Gamma \cdot \lambda x.S[T_1, \dots, T_n])U))$ par hypothèse, donc $S|_p = S|_q$ et $[\Omega, S]|_p = [\Omega, S]|_q$. Par définition d'une épine, le préfixe $\lambda x.S$ est inclus dans le squelette de $\lambda x.T$. Par convention de Barendregt locale appliquée à ce dernier, les occurrences de x dans S sont soit toutes libres dans S , c'est-à-dire toutes liées par la racine λx , soit toutes liées par des λ -abstractions de S . Donc $[\Omega, S]|_p$ et $[\Omega, S]|_q$ sont affectées de la même façon par la substitution de x dans $[\Omega, S]$, et finalement

$$[\Omega, S]^{x:=U}|_p = [\Omega, S]^{x:=U}|_q$$

\square

5.4 Correction et complétude causales

La présente section démontre que les étiquettes du λ -calcul faible étiqueté défini à la section 5.3.1 reflètent fidèlement la causalité dans ce système, au moyen des théorèmes 5.27 et 5.29 dits de correction causale et de complétude causale. Ces deux théorèmes assemblés assurent que l'ensemble \mathcal{N} des noms

qui contribuent à un nom donné Ω est exactement l'ensemble des noms des pas de réduction qu'il est nécessaire d'effectuer pour créer un pas de réduction de nom Ω à partir d'un terme initial.

Pour énoncer ces résultats formellement nous avons besoin de redéfinir formellement la notion de nécessité dans ce contexte. Ces définitions s'inspirent des définitions abstraites posées par Glauert et Khasidashvili [GK96] pour axiomatiser la nécessité d'un radical vis-à-vis d'un ensemble de termes et la contribution d'une famille de radicaux à une autre.

Définition 5.20 (Ensemble de pas de réduction nécessaire). *Soit \mathcal{T} un ensemble de λ -termes étiquetés. Un ensemble \mathcal{R} de pas de réduction est **nécessaire** pour \mathcal{T} si toute séquence de réduction ρ telle que $\text{src}(\rho)$ est un λ -terme étiqueté initial et $\text{tgt}(\rho) \in \mathcal{T}$ contient un pas de réduction de \mathcal{R} .*

Définition 5.21 (Antécédent nécessaire). *Soient Ω_1 et Ω_2 deux noms de pas de réduction. Notons \mathcal{R}_1 l'ensemble des pas de réduction dont le nom est Ω_1 et \mathcal{T}_2 l'ensemble des sources des pas de réduction dont le nom est Ω_2 . Le nom Ω_1 est une **antécédent nécessaire** du nom Ω_2 , noté $\Omega_1 \hookrightarrow^n \Omega_2$, si l'ensemble de pas de réduction \mathcal{R}_1 est nécessaire pour l'ensemble de termes \mathcal{T}_2 .*

Nous avons donc une notion de nécessité \hookrightarrow^n qui nous donne la relation de dépendance entre les ensembles de pas de réduction portant des noms donnés, à mettre en face d'une notion de contribution \hookrightarrow qui indique simplement un lien entre des séquences d'étiquettes. La *correction* et la *complétude* causales de notre système étiqueté sont l'égalité des relations \hookrightarrow^n et \hookrightarrow . Ces propriétés nous assurent que la relation \hookrightarrow permet de lire dans les étiquettes les vraies relations de dépendance que décrit \hookrightarrow^n .

Pour obtenir ce résultat nous utilisons des résultats intermédiaires sur le λ -calcul faible étiqueté et les noms de radicaux, dont les preuves peuvent être directement adaptées de celles données dans l'article d'origine [BLM07]. Des preuves de résultats équivalents seront données au chapitre 7 dans le cadre des CRS orthogonaux.

Lemme 5.22 (Contribution directe). *Pour tout pas de réduction $\rho : t \rightarrow t'$ de nom Ω , si ρ_c est un pas de réduction de source t' et de nom Ω_c qui est créé par ρ , alors $\Omega \hookrightarrow^d \Omega_c$.*

Lemme 5.23 (Stabilité des pas de réduction). *Pour tout pas de réduction $\rho : t \rightarrow t'$, si ρ_a est un pas de réduction de source t et de nom Ω , alors tout descendant de $\text{root}(\rho_a)$ est la position racine d'un pas de réduction de même nom Ω .*

Lemme 5.24 (Développements finis). *Soient t un λ -terme étiqueté et \mathcal{R} un ensemble de pas de réduction de source t . Alors les trois propriétés suivantes sont vérifiées :*

- Tous les développements de \mathcal{R} sont finis.
- Tous les développements complets de \mathcal{R} ont même cible.
- Tous les développements complets de \mathcal{R} définissent les mêmes relations de descendance et de résiduation.

La *correction causale*, c'est-à-dire l'inclusion de la relation sur les séquences d'étiquettes \hookrightarrow dans la relation de dépendance \hookrightarrow^n (théorème 5.27), est obtenue en suivant les origines des étiquettes et en vérifiant que ces dernières n'enregistrent pas de « fausses » contributions.

Lemme 5.25 (Origine des étiquettes). *Soit ρ une séquence de réduction dont la source est un λ -terme étiqueté initial. Si une étiquette de la forme $[\Omega]$ ou $[\Omega, \alpha]$ apparaît dans la cible de ρ , alors la séquence ρ contient un pas de réduction de nom Ω .*

Démonstration. Par récurrence sur la longueur de ρ , en remarquant qu'à chaque pas une étiquette de la forme $[\Omega]$ ou $[\Omega, \alpha]$ soit est créée par un pas de réduction Ω soit descend d'une étiquette identique. \square

Lemme 5.26 (Nécessité directe). *Soit ρ une séquence de réduction dont la source est un λ -terme étiqueté initial. Notons ρ un pas de réduction de source $t = \text{tgt}(\rho)$, et Ω le nom de ρ . Si Ω' est un nom de pas de réduction tel que $\Omega' \hookrightarrow^d \Omega$, alors ρ contient un pas de réduction de nom Ω' .*

Démonstration. Notons $\Omega = \omega_1 \dots \omega_n$. Par définition de \hookrightarrow^d il existe un i tel que $\omega_i = [\Omega']$ ou $\omega_i = [\Omega', \alpha]$. Donc par définition du nom d'un pas de réduction il existe une position $p \in \text{pos}(t)$ telle que $\tau_p(t) = [\Omega']$ ou $\tau_p(t) = [\Omega', \alpha]$. Alors par lemme 5.25 la séquence ρ contient un pas de réduction de nom Ω' . \square

Théorème 5.27 (Correction causale). *Soient deux noms de pas de réduction Ω_1 et Ω_2 tels que $\Omega_1 \hookrightarrow \Omega_2$. Alors $\Omega_1 \hookrightarrow^n \Omega_2$.*

Démonstration. Par récurrence sur la longueur d'une plus courte séquence $\Omega_1 \hookrightarrow^d \dots \hookrightarrow^d \Omega_2$.

- Si $\Omega_1 \hookrightarrow^d \Omega_2$, alors par lemme 5.26 toute séquence de réduction allant d'un terme initial à un terme source d'un pas de réduction de nom Ω_2 contient un pas de réduction de nom Ω_1 , c'est-à-dire $\Omega_1 \hookrightarrow^n \Omega_2$.
- Si $\Omega_1 \hookrightarrow^d \Omega_3 \hookrightarrow \Omega_2$ et $\Omega_3 \hookrightarrow^n \Omega_2$, alors de même $\Omega_1 \hookrightarrow^n \Omega_3$ et par transitivité $\Omega_1 \hookrightarrow^n \Omega_2$. \square

La *complétude causale*, c'est-à-dire l'inclusion de la relation de dépendance \hookrightarrow^n dans la relation sur les séquences d'étiquettes \hookrightarrow (théorème 5.29), est obtenue en démontrant que des étapes de réduction dont les noms ne sont pas comparables par \hookrightarrow peuvent être permutées.

Lemme 5.28 (Permutation). *Si $t_0 \Rightarrow_{\Omega_0} t_1 \Rightarrow_{\Omega} t_2$ et $\Omega_0 \not\prec \Omega$, alors l'un de ces deux cas est vérifié :*

- $t_0 \Rightarrow_{\Omega} t_2$.
- Il existe t'_1 tel que $t_0 \Rightarrow_{\Omega} t'_1 \Rightarrow_{\Omega_0} t_2$.

Démonstration. Comme $\Omega_0 \not\prec \Omega$, par contraposition du lemme 5.22 les pas de réduction de source t_1 et de nom Ω ont des ancêtres de source t_0 . En particulier $t_0 \Rightarrow_{\Omega_0} t_1 \Rightarrow_{\Omega} t_2$ est un développement complet de l'ensemble des pas de réduction de source t_1 et de nom Ω_0 ou Ω et par lemme 5.24 sa cible t_2 est également

la cible de tout autre développement complet, par exemple le développement consistant d'abord en la réduction parallèle du nom Ω suivie de celle du nom Ω_0 si ce dernier apparaît encore. \square

Théorème 5.29 (Complétude causale). *Soient deux noms de pas de réduction Ω_0 et Ω tels que $\Omega_0 \hookrightarrow^n \Omega$. Alors $\Omega_0 \hookrightarrow \Omega$.*

Démonstration. Considérons un nom de pas de réduction Ω , et $\bar{\mathcal{R}}$ l'ensemble des séquences de réductions parallèles étiquetées de longueur minimale ayant pour source un terme initial et terminant par une étape Ω . Il existe donc un n tel que toute séquence $\bar{\rho} \in \bar{\mathcal{R}}$ a la forme

$$\Rightarrow_{\Omega_1^e} \dots \Rightarrow_{\Omega_n^e} \Rightarrow \Omega$$

Notons $\bar{\mathcal{R}}^*$ l'ensemble des séquences $\bar{\rho} \in \bar{\mathcal{R}}$ pour lesquelles il existe au moins un $i \in \{1, \dots, n\}$ tel que $\Omega_i^e \not\hookrightarrow \Omega$.

Supposons que l'ensemble $\bar{\mathcal{R}}^*$ est non vide. Notons i le plus grand entier tel qu'il existe une séquence $\bar{\rho} \in \bar{\mathcal{R}}^*$ telle que $\Omega_i^e \not\hookrightarrow \Omega$. Notons $\bar{\rho}$ une séquence de $\bar{\mathcal{R}}^*$ telle que $\Omega_i^e \not\hookrightarrow \Omega$. La séquence $\bar{\rho}$ est de la forme

$$\dots \Rightarrow_{\Omega_i^e} \Rightarrow_{\Omega_{i+1}^e} \dots \Rightarrow_{\Omega_n^e} \Rightarrow \Omega$$

(où l'étape $\Rightarrow_{\Omega_{i+1}^e}$ peut être confondue avec l'étape $\Rightarrow_{\Omega_n^e}$ ou l'étape \Rightarrow_{Ω}).

Par définition de $\bar{\rho}$, nous avons $\Omega_{i+1}^e \hookrightarrow \Omega$. Comme $\Omega_i^e \not\hookrightarrow \Omega$ et par transitivité de \hookrightarrow nous en déduisons $\Omega_i^e \not\hookrightarrow \Omega_{i+1}^e$. Alors par lemme 5.28 nous pouvons construire l'une des deux séquences suivantes :

- Une séquence $\dots \Rightarrow_{\Omega_{i-1}^e} \Rightarrow_{\Omega_{i+1}^e} \dots \Rightarrow_{\Omega}$ strictement plus courte que $\bar{\rho}$, contredisant la minimalité de sa longueur parmi les séquences de réductions parallèles étiquetées s'achevant par \Rightarrow_{Ω} .
- Une séquence $\dots \Rightarrow_{\Omega_{i+1}^e} \Rightarrow_{\Omega_i^e} \dots \Rightarrow_{\Omega}$ avec $\Omega_i^e \not\hookrightarrow \Omega$, de même longueur que $\bar{\rho}$ mais contredisant la maximalité de i .

Ainsi, l'ensemble $\bar{\mathcal{R}}^*$ est nécessairement vide, et tout nom Ω_0 apparaissant dans une séquence de $\bar{\mathcal{R}}$ vérifie $\Omega_0 \hookrightarrow \Omega$.

Finalement, soit Ω_0 un nom tel que $\Omega_0 \hookrightarrow^n \Omega$. Par définition 5.21, toute séquence de réduction ρ dont la source est initiale et dont la cible $\text{tgt}(\rho)$ est source d'un pas de réduction ρ de nom Ω contient un pas de nom Ω_0 . C'est le cas en particulier de toute séquence développant une séquence de $\bar{\mathcal{R}}$. Donc $\Omega_0 \hookrightarrow \Omega$. \square

5.5 Discussion : variantes de forme

Le λ -calcul étiqueté défini dans ce chapitre présente un certain nombre de différences de forme avec les systèmes étiquetés des chapitres précédents, pour décrire finalement une structure très similaire.

5.5.1 Contribution et ordre SLS

Un ordre de contribution \hookrightarrow sur les séquences d'étiquettes a été introduit pour exprimer les propriétés causales de cet étiquetage. Cet ordre présente quelques ressemblances avec l'ordre \leq sur les étiquettes utilisé pour la définition d'un SLS. Notamment, le lemme 6.59 de causalité directe, qui assure que le nom d'un créateur contribue au nom de sa création, rappelle l'axiome *Progression* qui demande que l'étiquette racine du radical d'un pas de réduction ρ soit plus petite que les étiquettes de la zone d'effet de ρ .

Pour faire apparaître un autre point de ressemblance, regardons ces deux relations dans le cas où les séquences d'étiquettes concernées sont toutes de longueur 1 :

- Les critères $\Omega \hookrightarrow^d \Gamma_1[\Omega]\Gamma_2$ et $\Omega \hookrightarrow^d \Gamma_1[\Omega, \alpha]\Gamma_2$ deviennent respectivement $\omega \hookrightarrow^d [\omega]$ et $\omega \hookrightarrow^d [\omega, \alpha]$ (où ω désigne la séquence Ω en soulignant qu'elle est réduite à une seule étiquette).
- Les critères $\omega_1 \leq [\omega_1 \dots \omega_n]$ et $\omega_1 \leq [\omega_1 \dots \omega_n, \alpha]$ deviennent respectivement $\omega_1 \leq [\omega_1]$ et $\omega_1 \leq [\omega_1, \alpha]$.

Nous obtenons deux relations identiques.

Une différence entre les deux peut en revanche être observée au niveau des séquences d'étiquettes :

- \hookrightarrow peut regarder n'importe quelle étiquette d'une séquence, et nous avons par exemple à la fois $\Omega_1 \hookrightarrow^d [\Omega_1][\Omega_2]$ et $\Omega_2 \hookrightarrow^d [\Omega_1][\Omega_2]$.
- \leq se limite toujours à la première étiquette d'une séquence, et nous avons par exemple $\omega_1 \leq [\omega_1 \omega_2]$ mais *pas* $\omega_2 \leq [\omega_1 \omega_2]$.

Une fois cette observation faite, il devient tentant d'autoriser la comparaison $\omega_2 \leq [\omega_1 \omega_2]$, dans un rapprochement approximatif de \hookrightarrow et \leq . Cependant, l'ordre ainsi obtenu ne validerait pas l'axiome *Héritage*.

Exemple 5.30.

Considérons le pas de réduction

$$\alpha : @(\beta : \lambda x.x, T) \rightarrow [\alpha\beta] : T$$

où T est un terme étiqueté complet quelconque. Son radical est tout le terme $\alpha : @(\beta : \lambda x.x, T)$, et sa zone d'effet couvre uniquement la position de l'étiquette $[\alpha\beta]$. Supposons que $\beta \leq [\alpha\beta]$. Comme l'étiquette β est différente de l'étiquette racine α du radical, pour valider l'axiome *Héritage* il faudrait qu'existât une étiquette γ dans le radical telle que $\beta < \gamma$. Ce n'est pas le cas.

Une autre libéralisation tentante depuis longtemps est d'ajouter la comparaison $\alpha \leq [\omega, \alpha]$ à la définition de \leq . Cette variante est à éviter exactement pour la même raison.

5.5.2 Les positions des étiquettes

Dans les chapitres 2, 3, et 4, nous avons construit des étiquetages de systèmes de réécriture en faisant porter les étiquettes par les symboles des systèmes.

En conséquence, les symboles de la grammaire étiquetée étaient les symboles d'origine, éventuellement munis d'étiquettes. Également, nous avons une fonction de « désétiquetage » $(.)^*$ qui retirait toutes les étiquettes des symboles d'un terme étiqueté t , et renvoyait un terme t^* de la grammaire d'origine ayant exactement le même ensemble de positions que t .

Dans le cadre du λ -calcul faible étiqueté de ce chapitre en revanche, nous avons utilisé les étiquettes elles-mêmes comme des symboles ayant leurs propres positions, qui s'ajoutent à l'ensemble des positions du terme à étiqueter. Pour entrer formellement dans le cadre des *SLS*, nous avons donc considéré ces étiquettes-symboles comme portées par un symbole factice distingué ι , qui s'ajoute à la syntaxe d'origine du λ -calcul sans en modifier la dynamique. Ce genre d'ajout, appelé *expansion*, sera regardé de plus près au chapitre 7.

Une autre manière d'appréhender les positions des étiquettes est de partitionner les positions en deux ensembles :

1. Les positions des constructions syntaxiques du λ -calcul (variables, abstractions, applications) qui forment la "vraie" structure de l'arbre de syntaxe d'un terme.
2. Les positions des étiquettes, qui s'intercalent dans cette structure et peuvent être vues comme les positions des arêtes de l'arbre de syntaxe.

Cette vision très fine consistant à suivre à la fois positions des nœuds et des arêtes d'un terme est suggérée notamment par Vincent van Oostrom [Ter03, Chap. 8].

5.5.3 Lien avec la pleine paresse

Pour comparer le λ -calcul faible étiqueté de ce chapitre avec l'étiquetage du β -système parcimonieux pleinement paresseux \mathcal{P}_{BLM} présenté au chapitre 3, nous proposons une fonction $\chi(\cdot)$ traduisant le premier en le second.

Les mécanismes de base de la traduction sont les suivants :

- L'étiquette contiguë à une application ou une λ -abstraction lui est associée tandis que toutes les autres sont oubliées.
- La traduction d'une étiquette ne conserve que la première étiquette de chaque séquence.

Formellement :

$$\begin{aligned}
 \chi(x) &= x \\
 \chi(\alpha : T) &= \chi(T) \\
 \chi(\alpha : (T_1 T_2)) &= (\chi(T_1)\chi(T_2))^{\chi(\alpha)} \\
 \chi(\alpha : \lambda x.T) &= \lambda^{\chi(\alpha)} x. \chi(T) \\
 \\
 \chi(a) &= a && a \in L \\
 \chi([\Omega, \alpha]) &= [\chi(\Omega), \chi(\alpha)] \\
 \chi(\alpha_1 \dots \alpha_n) &= \chi(\alpha_1)
 \end{aligned}$$

Remarquons que la traduction n'est pas définie sur les étiquettes de la forme $[\Omega]$. Ceci tient compte d'un invariant du λ -calcul faible étiqueté que nous n'avons

pas énoncé : nous pouvons imposer que toute occurrence d'une étiquette $[\Omega]$ soit dans un sous-terme de la forme $[\Omega] : \alpha : X$. Autrement dit aucun sous-terme de la forme $[\Omega] : (T_1 T_2)$ ou $[\Omega] : \lambda x. T$ ne peut apparaître dans un terme au cours d'une séquence de réduction étiquetée. D'une part ceci est stable par réduction car les étiquettes de la forme $[\Omega]$ sont toujours placées devant un terme étiqueté complet, et d'autre part ceci implique qu'aucun nom de radical ne commence par une étiquette de la forme $[\Omega]$ puisque la première étiquette d'un nom de radical est l'étiquette du nœud application de ce radical.

Exemple 5.31.

Le diagramme suivant montre à gauche un pas de réduction dans le λ -calcul étiqueté de la section 5.3.1 et à droite un pas de réduction dans le λ -calcul étiqueté généré par le β -système parcimonieux \mathcal{P}_{BLM} de la section 3.2, ces deux pas de réduction étant mis en relation par la fonction $\chi(\cdot)$

$$\begin{array}{ccc}
 \zeta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, \kappa : y), T) & \xrightarrow{\chi(\cdot)} & @^\zeta(\lambda^\alpha x. \lambda^\beta y. @^\gamma(x, y), \chi(T)) \\
 \downarrow & & \downarrow \\
 [\zeta \alpha] : [\zeta \alpha, \beta] : \lambda y. [\zeta \alpha, \gamma] : @([\zeta \alpha, \eta] : T, \kappa : y) & \xrightarrow{\chi(\cdot)} & \lambda^{[\zeta, \beta]} y. @^{[\zeta, \gamma]}(\chi(T), y)
 \end{array}$$

Nous pouvons de plus vérifier que cette fonction définit une bisimulation entre les deux systèmes. Il reste ensuite à vérifier que les deux systèmes définissent des réductions partagées bisimilaires à partir de termes initiaux, ce que nous ne détaillerons pas ici. Le chapitre 7 discutera ce point dans le cadre plus général des *CRS*.

5.6 Discussion : causalités

Les sections 5.2 et 5.4 ont montré comment l'étiquetage décrit dans ce chapitre rendait fidèlement compte de la causalité dans le λ -calcul faible. Une justification intuitive de ces bonnes propriétés était le strict parallélisme entre les modifications des étiquettes et un critère de causalité directe exprimé en termes de création de radicaux.

Ce « strict parallélisme » fut un bien beau mensonge. Nous avons en fait déjà largement tiré parti des possibilités offertes par la réduction faible d'anticiper ou retarder les modifications d'étiquettes et donc la propagation des informations causales.

5.6.1 Des causes du dégel

Revenons sur le critère de création de radicaux lié à la réduction faible. Le principe en est : un pré-radical gelé par la présence d'une variable liée peut être dégelé par la substitution de cette variable. Nous en avons déduit *pour*

*faire bonne mesure*¹ qu'il convenait de marquer du nom d'un β -radical toutes les positions soumises à sa substitution.

Avec cette discipline, assurément nous touchons toutes les positions dégelées par la réduction. Mais comme le montre l'exemple 5.32 nous modifions aussi les étiquettes de positions qui restent gelées. Les marques que nous laissons dans les étiquettes d'un réduit vont donc au-delà des positions dégelées.

Exemple 5.32.

Considérons le terme non étiqueté

$$@(\lambda x. @(\lambda y. @(\lambda z. x, y), t), u)$$

où t et u sont des termes quelconques et son étiquetage

$$\alpha : @(\beta : @(\gamma : \lambda x. \delta : \lambda y. \zeta : @(\eta : \lambda z. x, y), T), U)$$

où T et U sont des étiquetages complets quelconques de t et u . Nous pouvons y voir un pré-radical gelé

$$\begin{aligned} & @(\lambda z. x, y) \\ & \zeta : @(\eta : \lambda z. x, y) \end{aligned}$$

et un vrai radical formé par la λ -abstraction étiquetée γ et l'application étiquetée β . La réduction de ce radical nous mène à :

$$\begin{aligned} & @(\lambda y. @(\lambda z. t, y), u) \\ \alpha : @([\beta\gamma] : [\beta\gamma, \delta] : \lambda y. [\beta\gamma, \zeta] : @([\beta\gamma, \eta] : \lambda z. T, y), U) \end{aligned}$$

Dans ce terme, le descendant $@(\lambda z. t, y)$ du pré-radical gelé déjà observé $@(\lambda z. x, y)$ est toujours gelé, mais ses étiquettes ont été modifiées en :

$$[\beta\gamma, \zeta] : @([\beta\gamma, \eta] : \lambda z. T, y)$$

Il porte donc maintenant la marque du radical de nom $\beta\gamma$, sans toutefois être déjà dégelé. Voyons maintenant l'étape de réduction suivante, du radical se trouvant à la racine du terme obtenu. Nous avons une application étiquetée par α et une λ -abstraction étiquetée par la séquence $[\beta\gamma][\beta\gamma, \delta]$ le nom est donc $\Omega = \alpha[\beta\gamma][\beta\gamma, \delta]$. Remarquons en particulier que $\beta\gamma \hookrightarrow \Omega$.

$$[\Omega] : [\Omega, [\beta\gamma, \zeta]] : @([\beta\gamma, \eta] : \lambda z. T, U)$$

Le pré-radical

$$[\Omega, [\beta\gamma, \zeta]] : @([\beta\gamma, \eta] : \lambda z. T, U)$$

est maintenant dégelé et devient un radical de nom $[\Omega, [\beta\gamma, \zeta]][\beta\gamma, \eta]$. Ce nom porte les marques des deux radicaux précédents, qui ont tous deux contribué au dégel en instanciant chacun une des variables liées du pré-radical d'origine.

1. Cette expression s'applique historiquement à une situation dans laquelle il est fait *plus* que nécessaire.

Il y a ici une discordance visible entre une notion de causalité qui a l'air intuitive, où nous pouvons considérer que les radicaux de noms $\beta\gamma$ et Ω participent tous deux de la même façon au dégel du radical final puisque chacun supprime l'une des variables, et l'analyse de création des radicaux, dans laquelle le radical de nom $\beta\gamma$ n'est jamais responsable direct de la création du radical final puisque la réduction du radical de nom Ω doit systématiquement être placée entre les deux.

5.6.2 Causalité retardée

Nous nous posons ici la question suivante : est-il possible de modifier l'étiquetage pour le faire mieux correspondre aux créations ?

La piste suivie est celle-ci : un radical est susceptible de dégeler des positions le long de l'épine de sa λ -abstraction principale, mais pas si ces positions font déjà partie de l'épine d'une autre λ -abstraction interne. Par conséquent, nous analysons la discipline qui consisterait à ne pas modifier les étiquettes des épines internes de la λ -abstraction principale. Formellement, un tel réétiquetage pourrait être défini par :

$$\begin{aligned} \Omega(x)(\alpha : Z) &= [\Omega, \alpha] : (\Omega(x)Z) \\ \Omega(x)x &= x \\ \Omega(x)(T_1 T_2) &= (\Omega(x)T_1)(\Omega(x)T_2) \\ \Omega(x)(\lambda y.T) &= \lambda y.S[\Omega(x)T_1, \dots, \Omega(x)T_n] \quad y \neq x, S = \langle T \rangle^{\{y\}}, T = S[T_1, \dots, T_n] \end{aligned}$$

Exemple 5.33.

Réduisons le radical

$$\zeta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y), T)$$

en utilisant le réétiquetage réduit suggéré dans cette section. Nous obtenons le terme

$$[\zeta \alpha] : [\zeta \alpha, \beta] : \lambda y. \gamma : @([\zeta \alpha, \eta] : T, y)$$

dans lequel l'étiquette γ n'a pas été modifiée car elle est contenue dans l'épine $\lambda y. \gamma : @(\square, y)$.

La première question sur cet étiquetage est la propagation d'informations causales suffisantes. À titre d'indice positif dans cette direction, nous pouvons remarquer dans l'exemple 5.32 une certaine redondance dans les informations. La contribution de $\beta\gamma$ est en effet marquée quatre fois :

- une directe dans $[\beta\gamma, \eta]$,
- deux indirectes à l'intérieur de $\Omega = \alpha[\beta\gamma][\beta\gamma, \delta]$, et
- une indirecte dans $[\beta\gamma, \zeta]$.

Le troisième cas est un artefact syntaxique, qui n'est pas vu par la clôture transitive de \multimap^d . De même le fait que $\beta\gamma$ apparaisse deux fois dans le deuxième cas n'a pas d'importance ici.

Les deux premiers cas en revanche (contribution directe et contribution par l'intermédiaire de Ω) ont une signification : ils soulignent que le radical de nom $\beta\gamma$ participe au dégel du radical final par deux phénomènes simultanés :

- Il fait disparaître l'une des variables responsables du gel.
- Il dégèle le radical de nom Ω qui sera le responsable le plus direct du dégel du radical final.

Ainsi nous avons une redondance des informations causales aux positions qui sont directement affectées à la fois par $\beta\gamma$ et par Ω , puisque ces positions enregistrent à la fois la contribution directe (substitution d'une variable) et la contribution indirecte (création du radical de nom Ω). Comme la contribution indirecte est nécessairement marquée par transitivité ($\beta\gamma$ crée Ω qui crée le radical final), nous aurions donc pu nous passer du marquage de la contribution directe de $\beta\gamma$ dans l'épine de la λ -abstraction principale du radical de nom Ω . En quelque sorte, nous pouvons considérer que cette contribution sera marquée indirectement avant le dégel du radical final, et donc qu'il est possible de *retarder* la propagation des informations.

Une deuxième question concerne la préservation de la propriété de partage dans le cadre de ce nouvel étiquetage. La réponse est cette fois clairement négative, comme l'atteste l'exemple 5.34.

Exemple 5.34.

Considérons le terme étiqueté

$$\delta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y), \zeta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y), T))$$

où T est un terme étiqueté complet quelconque. La λ -abstraction $\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y)$ a deux occurrences partagées dans notre terme (qui peut être obtenu en une étape à partir d'un terme initial).

Réduisons le radical

$$\zeta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y), T)$$

en utilisant le réétiquetage réduit suggéré dans cette section. Nous obtenons le terme

$$[\zeta\alpha] : [\zeta\alpha, \beta] : \lambda y. \gamma : @([\zeta\alpha, \eta] : T, y)$$

dans lequel l'étiquette γ n'a pas été modifiée car elle est contenue dans l'épine $\lambda y. \gamma : @(\square, y)$. Ce sous-terme pris isolément a l'air légitime, mais en le remettant dans son contexte nous obtenons le terme

$$\delta : @(\alpha : \lambda x. \beta : \lambda y. \gamma : @(\eta : x, y), [\zeta\alpha] : [\zeta\alpha, \beta] : \lambda y. \gamma : @([\zeta\alpha, \eta] : T, y))$$

dans lequel cohabitent les deux expressions $\gamma : @(\eta : x, y)$ et $\gamma : @([\zeta\alpha, \eta] : T, y)$. La propriété de partage est brisée.

La non-préservation de la propriété de partage prouve que cet étiquetage réduit ne vérifie pas les axiomes des *SLS*. Le point de rupture est le suivant : les axiomes *Zone d'effet* et *Progression*, quand il sont mis ensemble, imposent à la zone d'effet d'un radical d'être un préfixe du réduit. En particulier, les étiquettes sont modifiées dans une partie connexe de l'arbre syntaxique, ce qui n'est pas le cas dans l'exemple 5.34.

Vient alors la troisième question : pouvons-nous construire un système étiqueté régi par ces principes de retardement de la propagation de certaines informations causales (principes qui ont l'air compatibles avec un bon enregistrement des relations causales entre radicaux) et qui de plus soit un *SLS* ? Une piste pour cela serait d'appliquer une transformation de programme rendant connexe la zone d'effet du réétiquetage réduit. Le λ -*lifting* étudié au chapitre 8 aura cet effet.

5.6.3 Causalité anticipée

Revenons sur ce qui, en réduction faible, nous donne une certaine latitude quant à la propagation des informations causales. Les positions gelées sont en effet soumises à un double phénomène :

- Les étiquettes modifiées mais non dégelées, comme leur statut l'indique, restent gelées. En particulier elles n'interviennent dans le nom d'aucun radical, et sont donc en quelque sorte « invisibles ».
- Avant d'être dégelées, ces étiquettes marquées en surplus vont être modifiées à nouveau. À ce moment elles vont en particulier recevoir à nouveau la marque du premier radical. Comme nous pouvons le voir dans l'exemple 5.32, cette marque est en effet incluse dans le nom du deuxième radical.

Cette observation nous a permis de justifier ci-dessus la pertinence causale d'un étiquetage limitant la propagation des informations. Les effets étaient un retard dans la diffusion des informations (invisible car ne touchant que des positions gelées) ainsi qu'une diminution des redondances de ces informations.

Mais nous pouvons de même nous servir de ces observations dans l'autre sens, et cette fois accélérer la diffusion des informations causales. Ainsi toute position gelée qui, par sa situation, sera à coup sûr marquée indirectement par un certain nom Ω avant son dégel, peut être marquée directement dès le moment de la réduction du radical de nom Ω .

Nous vérifierons cette prédiction dans les chapitres à venir, dans lesquels la modification des étiquettes se fera non pas le long de l'épine uniquement, mais dans tout le squelette des abstractions considérées.

Remarquons également que les explications données ci-dessus à la préservation des propriétés causales malgré une certaine anticipation des contributions ressemblent beaucoup à l'explication donnée dans l'exemple 3.20 de la préservation de la réduction partagée malgré des duplications différentes de certains nœuds. C'est effectivement le même phénomène qui est ici décrit, et nous obtenons ici un nouvel éclairage sur l'équivalence des multiples approches de la pleine paresse.

5.7 Discussion : reconstruction

Afin de comparer et classer différentes techniques d'étiquetage pour les systèmes de réécriture, van Oostrom [Ter03, Chap. 8] a proposé deux critères

objectifs contre lesquels peuvent être testés les étiquetages : la *maximalité* et la *reconstruction*.

Le premier, la maximalité, est lié à une notion de projection qui permet d'exprimer qu'un étiquetage est plus général qu'un autre. La manière dont nous avons ci-dessus projeté l'étiquetage causal de ce chapitre sur un des étiquetages pleinement paresseux du chapitre 3 illustre bien cette idée.

Le deuxième, la reconstruction, permet d'exprimer qu'un système d'étiquetage fournit un bon enregistrement des événements de réduction, en ce qu'il permet de reconstruire les pas de réduction passés à permutations et à effacements près.

5.7.1 Reconstruction ambiguë

Les étiquettes présentées ici pour le λ -calcul faible ont été amenées comme un enregistrement distribué de l'histoire des pas de réductions. Nous avons même vérifié que cet enregistrement était suffisamment précis pour ne rien oublier des relations causales entre les différents radicaux.

Pourtant, ce λ -calcul étiqueté faible ne vérifie pas la propriété de reconstruction. En effet, s'il est aisé d'isoler les descendants de l'épine principale d'un radical dans son réduit, il est impossible de distinguer l'argument du radical des paramètres du corps de la fonction (exemple 5.35).

Exemple 5.35.

Considérons le terme

$$[\Omega] : [\Omega, \alpha] : (T_1([\Omega, \beta] : (T_1 T_2)))$$

où Ω est une certaine séquence $\gamma\delta$, où $\alpha, \beta, \gamma, \delta$ sont des étiquettes initiales, et où $T_1 T_2$ sont des termes étiquetés complets. L'histoire visible de ce terme est constituée d'une unique réduction, qui concerne un radical de nom $\Omega = \gamma\delta$. Ce radical était donc de la forme :

$$\gamma : ((\delta : \lambda z.U)V)$$

où U et V sont deux termes étiquetés par des étiquettes initiales. Nous pouvons également obtenir des informations sur l'épine de $\lambda z.U$. En effet, les positions portant des étiquettes de la forme $[\Omega, .]$ sont exactement les descendantes de cette épine. Nous reconstruisons donc :

$$U = \alpha : (U_1(\beta : (U_2 U_3)))$$

où U_1, U_2 et U_3 sont trois termes étiquetés complets dont chaque position vérifie l'une de ces conditions :

- Elle ne fait pas partie de la z -épine de U . C'est-à-dire que z n'y apparaît pas libre.
- Elle fait partie de la z -épine de U mais n'a pas de descendant. C'est le cas uniquement des occurrences libres de z .

Nous ne pouvons donc pas à ce stade dire qui de U_1 , U_2 ou U_3 est une occurrence de la variable z de la fonction ou un paramètre extérieur à l'épine de la fonction.

En d'autres termes nous ne savons pas qui des deux occurrences de T_1 ou de T_2 est l'argument de la réduction ou une partie du corps de la fonction. En effet, les sources suivantes sont toutes possibles :

$$\begin{aligned} \gamma &: ((\delta : \lambda z. \alpha : (T_1(\beta : (T_1 z))))T_2) \\ \gamma &: ((\delta : \lambda z. \alpha : (z(\beta : (z T_2))))T_1) \\ \gamma &: ((\delta : \lambda z. \alpha : (T_1(\beta : (z T_2))))T_1) \end{aligned}$$

Bilan : la reconstruction est une propriété plus forte que les propriétés causales présentées jusqu'ici.

5.7.2 Retrouver la propriété de reconstruction

En revanche, la version d'origine de ce calcul, par Blanc, Lévy et Maranget [BLM07], possède la propriété de reconstruction en plus de toutes les propriétés causales que nous avons vues dans ce chapitre. Observons la différence.

Dans la réduction d'un radical de nom Ω , le λ -calcul faible étiqueté de ce chapitre crée deux sortes d'étiquettes dans le réduit : une étiquette $[\Omega]$ à la racine, et une étiquette $[\Omega, \alpha]$ pour remplacer chaque étiquette α présente dans l'épine du radical. Ces deux cas suivent les deux cas (1) et (2⁺) de création de radicaux.

Le λ -calcul faible étiqueté original [BLM07] présente deux différences avec ce que nous avons vu dans ce chapitre :

- Les variables sont considérées comme des termes étiquetés incomplets, et doivent donc être étiquetées.
- Une troisième construction $[\Omega]$ est utilisée pour les étiquettes. Ces étiquettes $[\Omega]$ sont intercalées aux abords de chaque position où une variable est substituée. Ceci correspond au cas de création (2).

L'omission des étiquettes $[\Omega]$ est possible essentiellement car le cas de création (2) est couvert par le cas (2⁺). Ainsi, les informations causales d'une étiquette $[\Omega]$ sont redondantes avec celles contenues dans une certaine étiquette $[\Omega, \alpha]$ proche, par exemple l'étiquette de la variable x^α qui vient d'être substituée.

Exemple 5.36.

Considérons dans le λ -calcul étiqueté de Blanc, Lévy et Maranget [BLM07] le terme

$$\alpha : @(\beta : \lambda x. \gamma : @(\delta : x, T), \zeta : \lambda y. U)$$

où T et U sont deux termes étiquetés complets quelconques. Sa réduction dans le λ -calcul faible étiqueté original [BLM07] donne le terme

$$[\alpha\beta] : [\alpha\beta, \gamma] : @([\alpha\beta, \delta] : [\alpha\beta] : \zeta : \lambda y. U, T)$$

dans lequel l'étiquette $[\alpha\beta]$ est accolée à l'étiquette $[\alpha\beta, \delta]$.

La justification a priori de l'omission des étiquettes des variables est un peu plus subtile, et est liée à la définition des noms des radicaux. Rappelons-nous que l'étiquetage de Lévy du λ -calcul [Lé78] utilisait comme nom pour un radical la séquence d'étiquettes portée par la branche gauche du nœud d'application du radical, c'est-à-dire tout ce qui se trouvait entre l'application et l'abstraction. Cependant Blanc, Lévy et Maranget [BLM07] ont introduit un changement dans cette première technique et ont encore incorporé au nom d'un β -radical l'étiquette de son nœud d'application. C'est ce dernier exemple que nous suivons ici.

Regardons maintenant les cas de création (2) et (2⁺), qui sont les seuls qui traitent directement avec une variable : ils concernent des substitutions qui viennent d'au-dessus du radical dans son entier. En particulier ces substitutions laissent leur marque sur le nœud d'application du radical, ce qui nous dispense de marquer en plus les positions des variables substituées.

Exemple 5.37.

Considérons dans le λ -calcul étiqueté de Blanc, Lévy et Maranget [BLM07] le radical

$$[\alpha\beta, \gamma] : @([\alpha\beta, \delta] : [\alpha\beta] : \zeta : \lambda y.U, T)$$

qui a été créé par l'étape de réduction montrée à l'exemple 5.36 selon le cas de création (2). Son nom $[\alpha\beta, \gamma][\alpha\beta, \delta][\alpha\beta]\zeta$ fait apparaître plusieurs fois le contributeur $\alpha\beta$. L'occurrence $[\alpha\beta]$ discutée ci-dessus comme l'occurrence $[\alpha\beta, \delta]$ héritée de l'étiquette de la variable substituée peuvent être vues comme redondantes par rapport à l'occurrence $[\alpha\beta, \gamma]$ qui est de toute façon marquée par le réétiquetage de l'épine du radical réduit.

Ainsi, nous avons pu nous passer dans ce chapitre de la construction $[\Omega]$ utilisée par Blanc, Lévy et Maranget [BLM07] sans hypothéquer la correction ni la complétude causale. Toutefois, les étiquettes $[\Omega]$ apportent des informations précieuses à la reconstruction, puisque ce sont elles qui permettent de distinguer les occurrences substituées de l'argument des paramètres du corps de la fonction.

Exemple 5.38.

En reprenant les trois antécédents possibles de

$$[\Omega] : [\Omega, \alpha] : (T_1([\Omega, \beta] : (T_1 T_2)))$$

vus plus à l'exemple 5.35 nous obtenons en effet trois réductions différentes, dont les cibles ne peuvent plus être confondues :

$$\begin{aligned} \gamma : ((\delta : \lambda z. \alpha : (T_1(\beta : (T_1 z))))T_2) &\rightarrow [\Omega] : [\Omega, \alpha] : (T_1([\Omega, \beta] : (T_1([\Omega] : T_2)))) \\ \gamma : ((\delta : \lambda z. \alpha : (z(\beta : (z T_2))))T_1) &\rightarrow [\Omega] : [\Omega, \alpha] : (([\Omega] : T_1)([\Omega, \beta] : (([\Omega] : T_1)T_2))) \\ \gamma : ((\delta : \lambda z. \alpha : (T_1(\beta : (z T_2))))T_1) &\rightarrow [\Omega] : [\Omega, \alpha] : (T_1([\Omega, \beta] : (([\Omega] : T_1)T_2))) \end{aligned}$$

Ainsi, la présence des étiquettes $[\Omega]$ dans le λ -calcul faible étiqueté de Blanc, Lévy et Maranget [BLM07] permet de lever l'ambiguïté mise en lumière

par l'exemple 5.35. Pour défaire un pas de β -réduction étiquetée nous pouvons d'abord utiliser une étiquette $[\Omega]$ pour marquer la présence d'un réduct, puis utiliser les étiquettes de la forme $[\Omega, \alpha]$ pour trouver le squelette de la λ -abstraction principale du β -radical, et enfin utiliser les étiquettes $[\Omega]$ pour distinguer l'argument du β -radical des expressions libres de la λ -abstraction principale.

Le seul défaut de reconstruction est le cas général dans lequel l'argument du β -radical est effacé, dans lequel évidemment ledit argument n'est pas récupérable. Ce cas est prévu par van Oostrom [Ter03], qui limite en fait la reconstruction aux pas non effaçables d'une séquence de réduction.

5.8 Discussion : une autre preuve de partage

La preuve que donnent Blanc, Lévy et Maranget de la préservation de la propriété de partage pour le λ -calcul étiqueté faible [BLM07] a été inspiratrice des *SLS* définis dans cette thèse, et fournit la structure de la preuve du théorème 1.19 de partage. À ce stade nous pouvons cependant noter une différence importante entre les deux preuves : celle de Blanc, Lévy et Maranget repose sur les propriétés causales de l'étiquetage, alors que les *SLS* s'en affranchissent. Cette différence est une des contributions techniques majeures des *SLS*, et la clé de leur expressivité.

5.8.1 Une preuve causale du partage

La preuve de préservation de la propriété de partage de Blanc, Lévy et Maranget, comme celle présentée ici, peut se réduire à deux arguments :

- Les sous-termes modifiés par une réduction le sont d'une manière cohérente. Ceci correspond à l'axiome *Partage*.
- Aucune collision n'est possible entre un sous-terme modifié et un sous-terme non-modifié.

Dans la preuve de Blanc, Lévy et Maranget ce dernier fait est déduit d'un invariant de *maximalité* sur les noms de radicaux, tandis que la preuve des *SLS* utilise un invariant d'*indépendance* sur les étiquettes.

Le raisonnement sur la maximalité des noms de radicaux est le suivant : comme toutes les étiquettes créées par une étape de réduction incluent le nom Ω du radical réduct par cette réduction, une collision ne peut advenir que si une étiquette incluant le nom Ω du radical à réduire est déjà présente avant cette réduction. Il suffit donc pour empêcher ceci d'un invariant exprimant que les noms des radicaux d'un terme t sont *maximaux*, dans le sens qu'ils ne sont inclus dans aucune étiquette de t . Pour assurer la préservation de cet invariant, ils utilisent alors deux propriétés des noms des radicaux de leur système :

- tout résidu porte le même nom que son ancêtre, et
- tout radical créé porte un nom strictement « plus grand » que le nom de son créateur.

Nous pouvons reconnaître dans ces deux points deux des propriétés causales du λ -calcul faible étiqueté, à savoir les lemmes 5.23 de stabilité des radicaux et 5.22 de contribution directe.

5.8.2 Éloignement de la causalité

La question alors posée est la suivante : ces propriétés causales qui semblent utiles à la préservation de bonnes propriétés de partage sont-elles nécessaires ? Un des mérites du cadre axiomatique des *SLS* est de montrer qu'à défaut d'être totalement abandonnées, ces propriétés causales peuvent être significativement affaiblies.

Remarquons d'abord que les *SLS*, comme le λ -calcul faible étiqueté, garantissent l'absence de collisions par une certaine croissance forcée des étiquettes au cours de la réduction. Cette croissance est exprimée par le lemme 5.22 de contribution directe pour le λ -calcul faible étiqueté, et par l'axiome *Progression* pour les *SLS*.

La différence tient ensuite dans le fait que, en abandonnant l'usage de noms riches pour les radicaux, la propriété de stabilité de ces derniers est vidée de sa substance. Deux choses peuvent en effet advenir sans mettre en danger la preuve de partage des *SLS* :

- Une étiquette qui participe au nom d'un radical dans l'approche causale mais n'est pas la racine du radical est modifiée. Alors la stabilité des noms ne tient pas, mais les *SLS* ne voient pas la différence. En effet, nous pouvons constater à ce moment que toutes les occurrences du radical observent le même changement, et restent donc bien un seul radical partagé.
- L'étiquette racine du radical est modifiée. Le radical change alors d'identité. Éventuellement, si plusieurs occurrences partagées de ce radical existaient, elles peuvent être séparées en deux familles.

Ainsi les propriétés de causalité les plus fortes, qui sont vérifiées par le λ -calcul mais dont nous verrons au chapitre 7 qu'elles dépendent de l'orthogonalité des systèmes, ne sont pas utiles à la définition du partage dans les *SLS*, ce qui va nous permettre dans les chapitres à venir d'observer des étiquetages de systèmes pour lesquels la notion de causalité est plus complexe.

Bilan du chapitre

Dans ce chapitre nous avons analysé la structure causale du λ -calcul faible et détaillé comment cette dernière pouvait être fidèlement représentée dans un étiquetage du λ -calcul. Pour ceci nous avons caractérisé les manières dont un nouveau β -radical pouvait être créé par un pas de réduction (lemme 5.4) et en avons déduit une manière de propager les informations causales via des étiquettes. Nous avons également pu démontrer que les informations causales visibles dans les étiquettes correspondaient exactement aux relations de dépendances entre familles de radicaux de même nom (théorèmes 5.27 et 5.29 de

correction et de complétude causale).

Nous avons vu également que cet étiquetage causal définissait un *SLS* dont il sera prouvé qu'il spécifie un partage pleinement paresseux.

Cette approche en deux parties, avec d'abord une analyse des mécanismes de création de radicaux puis une propagation des informations dans un système étiqueté, donne la structure des deux chapitres suivants, qui reproduisent de telles analyses dans des cadres plus riches :

- Le chapitre 6 concerne un autre cas particulier plus compliqué (le *Pure Pattern Calculus*).
- Le chapitre 7 plonge dans le cadre général des *CRS*.

CHAPITRE 6

Histoire d'un calcul non-séquentiel

Sommaire

6.1 Filtrage et motifs dynamiques	178
6.2 <i>Pure Pattern Calculus</i>	181
6.3 Causalité dans le <i>Pure Pattern Calculus</i>	187
6.4 Filtrage étiqueté	192
6.5 <i>Pure Pattern Calculus</i> faible étiqueté	199
6.6 Propriétés causales	203
6.7 Discussion : non-stabilité	215

Nous allons dans le présent chapitre tenter de reproduire dans un cadre plus complexe les analyses de causalité effectuées au chapitre 5 sur le λ -calcul faible. Le formalisme considéré ici est le *Pure Pattern Calculus* (PPC) de Jay et Kesner [JK09], un calcul d'ordre supérieur intégrant de puissants mécanismes de filtrage.

Ce cadre présente deux particularités par rapport au simple λ -calcul :

- Les pas de réduction du *Pure Pattern Calculus* sont exprimés par l'entremise d'opérations de filtrage effectuées au niveau « méta », de sorte que ce qui se passe réellement lors d'un pas de réduction n'est pas directement et entièrement visible dans la règle.
- Le *Pure Pattern Calculus* est « non-séquentiel », dans le sens qu'il n'est pas toujours possible de désigner un pas de réduction qui soit nécessaire à l'obtention d'une forme normale. Et cette non-séquentialité n'est qu'une des manifestations, la plus visible, de ce qui fait la richesse des comportements du calcul. Ainsi le point de vue de la causalité révélera également une forme de non-orthogonalité du *Pure Pattern Calculus*, le défaut de *stabilité*, qui amène à réfléchir sur la notion même de nécessité.

La section 6.1 motive, du point de vue de la programmation, le cadre du *Pure Pattern Calculus* dont le formalisme est rappelé à la section 6.2. La section 6.3 analyse les cas dans lesquels un pas de réduction peut créer un radical dans

ce calcul. En suivant cette analyse, la section 6.4 fixe l'interaction des méta-opérations de filtrage avec les étiquettes, et la section 6.5 achève la mise en place d'un étiquetage du *Pure Pattern Calculus* faible définissant un *SLS*. Enfin, la section 6.6 établit les propriétés causales vérifiées par le système étiqueté, et la section 6.7 discute des propriétés causales qui ne sont pas vérifiées et des réflexions qu'elles imposent sur la notion de nécessité.

6.1 Filtrage et motifs dynamiques

Le filtrage peut être vu comme un mécanisme dont l'objet est de permettre la définition d'une fonction par cas sur la structure de son argument. Il est pour le programmeur une caractéristique très appréciable du paradigme fonctionnel. Cependant les mécanismes de filtrage usuels, bien que pratiques, souffrent d'un certain manque de généralité illustré par les exemples ci-dessous.

Exemple 6.1.

Considérons une structure de données représentant des arbres binaires : un arbre est soit une feuille portant une donnée, soit un nœud composé de deux sous-arbres. On peut l'écrire à la ML comme suit :

```
type 'a tree =
  | Data of 'a
  | Node of 'a tree * 'a tree
```

Une fonction *map* mettant à jour toutes les données contenues par un arbre binaire s'écrit facilement par cas sur la structure inductive des arbres, à l'aide du filtrage.

```
let rec map f t = match t with
  | Data a   -> Data (f a)
  | Node l r -> Node (map f l) (map f r)
```

Dans l'exemple 6.1, lorsque la fonction *map* est appliquée à deux arguments *f* et *t*, ce dernier est comparé à la forme *Data a*, appelée *motif*. En cas de succès (le motif et l'argument correspondent, soit $t = \text{Data } u$ pour un certain objet *u*), la *variable de filtrage* *a* capture la sous-structure *u* de *t*, et la fonction retourne la donnée mise à jour *Data (f(u))*. Si la première comparaison échoue (le motif et l'argument ne correspondent pas), alors l'*alternative* *Node l r* est testée à son tour.

Le code de *map* nomme explicitement tous les constructeurs qui peuvent être rencontrés lors de l'évaluation, c'est-à-dire *Data* et *Node*. Ainsi, toute variation même mineure sur la structure de données (arbres d'arité différente ou variable, données hétérogènes. . .) requiert l'écriture d'une nouvelle fonction *map*.

Une première solution à ce problème est donnée par le *polymorphisme de chemin* [Jay04, JK09], qui permet à une fonction générique de traverser récursivement n'importe quelle structure de données pour agir sur certains cas de base fixés (comme *Data a* ici). Le polymorphisme de chemin peut être réalisé par une classification universelle très simple des structures : chaque structure

est soit atomique, soit composée. Une structure atomique (appelée un *atome*), comme un constructeur isolé, est inerte. Une structure composée (appelée un *composé*), comme un constructeur (partiellement) appliqué, mène à la récursion.

Exemple 6.2.

Le code ci-dessous montre à quoi pourrait ressembler une version de map avec polymorphisme de chemin

```
let rec map f t = match t with
| Data a -> Data (f a)           //Base
| x y   -> (map f x) (map f y)   //Composé
| z     -> z                     //Atome
```

Et voici un exemple d'évaluation de cette fonction, où ++ est la fonction successeur, où chaque ligne est une étape de calcul, et où les crochets sont utilisés comme parenthèses pour séparer les appels de fonctions.

```
map ++ (Node (Data 1) (Data 2))
[map ++ (Node (Data 1))] [map ++ (Data 2)]
[map ++ Node] [map ++ (Data 1)] [map ++ (Data 2)]
Node (Data 2) (Data 3)
```

Alors que Node est un atome, des termes comme Node (Data 1) et Node (Data 1) (Data 2) sont composés.

Si maintenant nous considérons une nouvelle structure comme une liste

```
type 'a list =
| Nil
| Cons of 'a * 'a list
```

alors nous pouvons toujours utiliser la fonction map sur une liste de données

```
map ++ (Cons (Data 3)(Cons (Data 4) Nil))
[map ++ (Cons (Data 3))] [map ++ (Cons (Data 4) Nil)]
[map ++ Cons] [map ++ (Data 3)] ([map ++ (Cons (Data 4))] [map ++ Nil])
[map ++ Cons] [map ++ (Data 3)] ([map ++ Cons] [map ++ (Data 4)] [map ++ Nil])
Cons (Data 4) ( Cons (Data 5) Nil )
```

où cette fois Cons et Nil sont des atomes tandis que Cons (Data 3) par exemple est un composé.

Cette décomposition des structures en composés et atomes peut être encodée dans les langages usuels en utilisant un constructeur explicite pour démarquer les composés. Remarquons toutefois que cette solution appliquée brutalement malmène le typage des structures. Une conciliation entre typage et polymorphisme de chemin existe par exemple avec Generic Haskell [HJ03], qui intègre un mécanisme de raisonnement par cas sur le type d'une structure de données.

Comme nous le montre notre version de map de l'exemple 6.2, certains constructeurs (comme Node) peuvent être traités automatiquement grâce au polymorphisme de chemin, mais d'autres (comme Data) ont toujours besoin d'être mentionnés explicitement dans le code. Ceci limite le polymorphisme : si

nous voulons traiter une nouvelle structure dans laquelle certaines données sont représentées par quoi que ce soit d'autre que ce fameux constructeur `Data`, par exemple un arbre dont les feuilles sont marquées d'un constructeur `Leaf`, alors il nous faut encore écrire une nouvelle version de `map`, comme dans l'exemple 6.3.

Exemple 6.3.

Une nouvelle version de `map` avec polymorphisme de chemin, spécialisée pour les données de forme `Leaf a`.

```
let rec map f t = match t with
  | Leaf a -> Leaf (f a)           //Base
  | x y    -> (map f x) (map f y)  //Composé
  | z      -> z                    //Atome
```

Une deuxième solution, qui résout ce dernier problème, est donnée par le *polymorphisme de motif* [JK09, Jay09], qui introduit des motifs paramétrés. Ceci permet d'instancier un même motif de différentes manières. Ainsi, l'hypothétique fonction décrite à l'exemple 6.4 pourrait s'instancier en les fonctions `map` de l'exemple 6.2 ou de l'exemple 6.3 si l'un des constructeurs `Data` ou `Leaf` lui était donné en paramètre.

Exemple 6.4.

Une nouvelle version de `map` avec à la fois polymorphisme de chemin et de motif prendrait un paramètre additionnel `c` pour construire le motif de son cas de base :

```
let rec map c f t = match t with
  | {a}   c a -> c (f a)           //Base
  | {x,y} x y -> (map c f x) (map c f y) //Composé
  | {z}   z   -> z                 //Atome
```

Ici, chaque cas précise de plus une liste de noms servant à discriminer les variables de filtrage des paramètres de motif.

Remarquons que l'application partielle de la fonction `map` de l'exemple 6.4 au constructeur `Data` donne exactement la version de `map` de l'exemple 6.2, et que l'application partielle de la fonction `map` de l'exemple 6.4 au constructeur `Leaf` donne exactement la version de `map` de l'exemple 6.3. Mais maintenant, d'autres constructeurs ou d'autres structures peuvent être utilisés pour spécifier les éléments qui doivent être mis à jour.

Pour faire intervenir en paramètre des structures complexes, un cas plus général et intéressant est celui où le paramètre d'un motif est une *fonction construisant un motif*. Une telle fonction doit nécessairement être évaluée avant que le filtrage soit résolu, et le motif devient ainsi *dynamique*. Ce dernier point est tout simplement hors de portée des langages de programmation usuels.

Les motifs dynamiques semblent être une clé dans l'élaboration de paradigmes de filtrage expressifs, mais ils sont hélas incompatibles avec les implémentations usuelles. En effet, les définitions par cas utilisant du filtrage sont normalement compilées en une série de tests de bas niveau organisés dans ce qui

est appelé génériquement un *automate de filtrage*. Étant donné qu'un automate de filtrage naïf a tendance à introduire de nombreux tests redondants ou autres inefficacités, les ensembles de motifs sont analysés afin d'obtenir une organisation optimisée qui minimise le coût moyen d'une opération de filtrage et opère un partage des tests redondants. Voir par exemple [FM01, Mar08]. Cependant, les motifs dynamiques sont par nature indéfinis au moment de la compilation, ce qui rend ce genre d'analyse statique impossible. De nouveaux mécanismes doivent donc être définis pour reconquérir certaines de ces optimisations perdues ou en découvrir de nouvelles. Note *petit'historique* : c'est face à ce problème que j'ai commencé à m'intéresser au partage dans les premiers temps de mon doctorat.

6.2 Pure Pattern Calculus

Le *Pure Pattern Calculus* (PPC) de Jay et Kesner [JK09]¹ est une généralisation du λ -calcul supportant naturellement la manipulation de structures de données inductives et le filtrage avec polymorphismes de chemin et de motif. Une des caractéristiques de ce calcul est qu'il fait des motifs des objets de première classe. Les motifs peuvent également être évalués et sont donc des *motifs dynamiques*. De plus, tout terme est *a priori* susceptible d'être utilisé comme motif : c'est l'opération de filtrage elle-même qui est chargée de rejeter dynamiquement les utilisations « inappropriées ». La présente section rappelle le formalisme de ce calcul.

Définition 6.5 (Sous-séquences). Soient $\Gamma = \gamma_1 \dots \gamma_n$ et $\Delta = \delta_1 \dots \delta_k$ deux séquences d'éléments quelconques. Nous disons que Γ est une **sous-séquence** de Δ , noté $\Gamma \sqsubseteq \Delta$, s'il existe une fonction $\varphi : \{1 \dots n\} \rightarrow \{1 \dots k\}$ strictement croissante telle que $\Gamma = \delta_{\varphi(1)} \dots \delta_{\varphi(n)}$.

Exemple 6.6.

Ces deux exemples rappellent que les sous-séquences ne sont pas forcément faites d'éléments consécutifs, mais qu'elles doivent en revanche respecter l'ordre :

$$\begin{array}{l} \beta\delta\zeta \sqsubseteq \alpha\beta\gamma\delta\zeta \\ yx \not\sqsubseteq xyz \end{array}$$

Convention 6.7. Si $\Gamma = \gamma_1 \dots \gamma_n$ est une séquence, par abus de notation nous utiliserons encore Γ dans des expressions ensemblistes pour désigner l'ensemble d'éléments $\{\gamma_1, \dots, \gamma_n\}$.

La syntaxe du *Pure Pattern Calculus* utilise remplace la λ -abstraction $\lambda x.t$ du λ -calcul, qui abstrait sur une variable, par une abstraction plus générale de

1. Nous prendrons comme référence la version publiée communément par Jay et Kesner en 2009 [JK09], qui est la plus expressive, et dont la non-séquentialité sera le piment de ce chapitre. Leur première version du *Pure Pattern Calculus* [JK06] souffrait de la présence d'exemples pathologiques et était séquentielle. La version utilisée par Jay dans son livre [Jay09] est essentiellement identique à notre référence [JK09], à ceci près qu'elle n'est pas séquentielle.

forme $\langle\theta\rangle t.t$, qui abstrait sur un terme (le motif). L'« application par juxtaposition » du λ -calcul est toujours utilisée, et sert à la fois à désigner l'application d'une fonction à un argument et l'application d'un constructeur à une donnée. Une catégorie syntaxique caractérise les termes qui représentent des structures de données, et une catégorie syntaxique dite des *formes testables* (*matchable forms* en version originale [JK09]) caractérise des termes dont la forme superficielle est stable et peut donc être testée par une opération de filtrage.

Définition 6.8 (Syntaxe de PPC).

$$\begin{array}{lll} a, b, p, r, s, t & ::= & x \mid \hat{x} \mid tt \mid \langle\theta\rangle t.t & \text{Termes} \\ d & ::= & \hat{x} \mid dt & \text{Structures de données} \\ m & ::= & d \mid \langle\theta\rangle t.t & \text{Formes testables} \end{array}$$

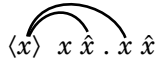
où $x, y, z \in \mathcal{X}$ sont des variables et θ, τ des séquences de variables.

Si précédemment nous avons utilisé la lettre θ pour dénoter des ensembles de variables, il est important de noter que nous considérons dans ce chapitre des séquences de variables, comme le requiert la définition originale de PPC.

Le terme $\langle\theta\rangle p.b$ est appelé un **cas**. La lettre p indique un terme utilisé comme motif (*pattern*) et b un terme utilisé comme corps de fonction (*function body*).

Dans ce cadre, une variable x peut avoir deux sortes d'occurrences : les occurrences *propres* (ou **variables**) x qui sont des occurrences ordinaires de variables pouvant être substituées, et les occurrences **matchables** \hat{x} qui ne peuvent pas l'être. Une *matchable* \hat{x} est une constante portant un nom x : elle peut être utilisée soit comme un constructeur soit comme une variable de filtrage (qui capture ce qui va être substitué à x), en fonction de son statut libre ou lié.

Comme illustré ci-dessous, dans un terme $\langle\theta\rangle p.b$ la séquence de variables θ lie les occurrences *matchables* dans p et les occurrences variables dans b .



Définition 6.9 (Liaisons). Les **variables libres** $\text{fv}(t)$ (free variables) et les **matchables libres** $\text{fm}(t)$ (free matchables) d'un terme t sont définies par les équations suivantes.

$$\begin{array}{ll} \text{fv}(x) & := \{x\} & \text{fm}(x) & := \emptyset \\ \text{fv}(\hat{x}) & := \emptyset & \text{fm}(\hat{x}) & := \{x\} \\ \text{fv}(t_1 t_2) & := \text{fv}(t_1) \cup \text{fv}(t_2) & \text{fm}(t_1 t_2) & := \text{fm}(t_1) \cup \text{fm}(t_2) \\ \text{fv}(\langle\theta\rangle p.b) & := \text{fv}(p) \cup (\text{fv}(b) \setminus \theta) & \text{fm}(\langle\theta\rangle p.b) & := (\text{fm}(p) \setminus \theta) \cup \text{fm}(b) \end{array}$$

Les **noms libres** $\text{fn}(t)$ (free names) d'un terme t sont définis par $\text{fn}(t) = \text{fv}(t) \cup \text{fm}(t)$, et ses **noms liés** $\text{bn}(t)$ (bound names) par les équations suivantes.

$$\begin{array}{ll} \text{bn}(x) & := \emptyset \\ \text{bn}(\hat{x}) & := \emptyset \\ \text{bn}(t_1 t_2) & := \text{bn}(t_1) \cup \text{bn}(t_2) \\ \text{bn}(\langle\theta\rangle p.b) & := \theta \cup \text{bn}(p) \cup \text{bn}(b) \end{array}$$

Une notion de renommage de variables ainsi qu'une notion de substitution sont déduites de ces critères de liaison.

Définition 6.10 (Renommage). *Pour tout $x \in \theta$ et tout y frais :*

$$\langle \theta \rangle p.b =_{\alpha} \langle \theta^{x:=y} \rangle p^{\{\hat{x}:=\hat{y}\}}.b^{\{x:=y\}}$$

Exemple 6.11.

$$\langle y \rangle \hat{x}y\hat{y}.xy\hat{y} =_{\alpha} \langle z \rangle \hat{x}y\hat{z}.xz\hat{y}$$

Dans le motif $\hat{x}y\hat{y}$, la matchable \hat{y} est liée et utilisée en tant que variable de filtrage, tandis que la matchable \hat{x} est libre et vue comme un constructeur. L'occurrence de variable y du motif est une variable libre, c'est-à-dire désigne paramètre extérieur, tandis que l'occurrence de variable y du corps est liée.

Définition 6.12 (Substitution). *Comme d'habitude, la substitution est une méta-opération définie par des équations.*

$$\begin{aligned} x^{\sigma} &= \sigma(x) & x \in \text{dom}(\sigma) \\ x^{\sigma} &= x & x \notin \text{dom}(\sigma) \\ \hat{x}^{\sigma} &= \hat{x} \\ (t_1 t_2)^{\sigma} &= t_1^{\sigma} t_2^{\sigma} \\ (\langle \theta \rangle p.b)^{\sigma} &= \langle \theta \rangle p^{\sigma}.b^{\sigma} & \theta \# \sigma \end{aligned}$$

La condition $\theta \# \sigma$ (" θ évite σ ") a pour fin d'éviter la capture de noms. La notation $\theta \# \sigma$ signifie $\theta \cap (\text{dom}(\sigma) \cup \text{fn}(\text{cod}(\sigma))) = \emptyset$, où $\text{dom}(\sigma)$ (resp. $\text{cod}(\sigma)$) dénote le domaine (resp. codomaine) de la substitution σ . La condition $\theta \cap \text{dom}(\sigma) = \emptyset$ assure qu'aucune occurrence de variable liée n'est substituée, et $\theta \cap \text{fv}(\text{cod}(\sigma)) = \emptyset$ (resp. $\theta \cap \text{fm}(\text{cod}(\sigma)) = \emptyset$) assure qu'aucune occurrence de variable (resp. matchable) de σ n'est capturée par θ dans b^{σ} (resp. p^{σ}). Pour les termes, $\theta \# t$ signifie $\theta \cap \text{fn}(t) = \emptyset$.

Le résultat d'une opération de filtrage est appelé un *filtrat* et noté μ . Il peut s'agir soit d'une substitution en cas de filtrage réussi, soit du symbole d'erreur \perp en cas d'échec de filtrage. Nous utiliserons de plus le symbole Σ pour dénoter les cas où l'opération de filtrage ne peut pas être menée jusqu'au bout en raison de la forme inappropriée du motif ou de l'argument. Cette « forme inappropriée » est caractérisée par la non-testabilité d'un sous-terme que l'opération de filtrage voudrait inspecter.

Définition 6.13 (Filtrat). *Les **filtrats définis** sont donnés par l'une de ces deux possibilités*

$$\mu ::= \sigma \mid \perp$$

où σ désigne une substitution de termes. Les **filtrats étendus** ajoutent le symbole « sablier » Σ aux filtrats définis

$$\mu^* ::= \mu \mid \Sigma$$

Le résultat du *filtrage* d'un argument a par un motif p avec un ensemble θ de variables de filtrage est noté $\{a/\theta p\}$ dans le *Pure Pattern Calculus* et est défini à partir d'une notion de filtrage syntaxique, appelée *filtrage composé*, caractérisée en ce qu'elle limite l'exploration récursive du motif et de l'argument aux applications qui sont des formes testables (aussi appelées *termes composés*).

Définition 6.14 (Filtrage composé). *Le **filtrage composé** d'un argument a par un motif p avec variables de filtrage θ , noté $\{a/\theta p\}$, est défini par les équations suivantes prises dans l'ordre :*

$$\begin{aligned} \{a/\theta \hat{x}\} &= \{x := a\} && x \in \theta \\ \{\hat{x}/\theta \hat{x}\} &= \{\} && x \notin \theta \\ \{a_1 a_2 / \theta p_1 p_2\} &= \{a_1 / \theta p_1\} \uplus \{a_2 / \theta p_2\} && a_1 a_2 \text{ et } p_1 p_2 \text{ formes testables} \\ \{a/\theta p\} &= \perp && a \text{ et } p \text{ formes testables, sinon} \\ \{a/\theta p\} &= \bar{X} && \text{sinon} \end{aligned}$$

où l'opérateur \uplus dénote l'union disjointe de filtrats et est défini par la table suivante :

\uplus	σ_2	\perp	\bar{X}
σ_1	$\sigma_1 \uplus \sigma_2$	\perp	\bar{X}
\perp	\perp	\perp	\perp
\bar{X}	\bar{X}	\perp	\bar{X}

où l'opérateur \uplus appliqué à des substitutions dénote leur union disjointe. Ainsi, $\sigma_1 \uplus \sigma_2 = \perp$ si $\text{dom}(\sigma_1) \cap \text{dom}(\sigma_2) \neq \emptyset$.

L'utilisation d'une union disjointe implique qu'un filtrage ne peut réussir si son motif n'est pas linéaire. Il s'agit d'un prérequis connu pour la confluence [Klo80].

Dans le cas du filtrat indéfini \bar{X} , le motif ou l'argument doit encore être évalué ou instancié pour permettre le filtrage. Tandis qu'assez naturellement une branche indéfinie \bar{X} l'emporte sur une branche générant une substitution σ , notons enfin qu'une branche générant un échec \perp l'emporte sur une branche indéfinie \bar{X} . Par ceci le filtrage du *Pure Pattern Calculus* n'est pas *séquentialisable*. Nous reviendrons sur ce point à la section 6.7.

Exemple 6.15.

Soit \hat{c} un constructeur.

- $\{\hat{c}/_y x \hat{y}\} = \bar{X}$, car le motif $x \hat{y}$ commence par une occurrence de variable et n'est pas une forme testable.
- $\{\hat{c}/_y (\langle z \rangle \hat{z}.z \hat{c}) \hat{y}\} = \bar{X}$, car le motif est maintenant un pré-radical (définition ci-dessous), qui n'est toujours pas une forme testable.
- $\{\hat{c}/_y \hat{y} \hat{c}\} = \perp$, car l'atome \hat{c} ne peut correspondre au composé $\hat{y} \hat{c}$.

Un exemple de filtrage composé réussi est donné par le filtrat $\{\hat{c}(\langle z \rangle \hat{z}.z \hat{c}) /_{x_1 x_2} \hat{x}_1 \hat{x}_2\}$: l'argument et le motif sont des formes testables, et chacune des variables de filtrage \hat{x}_1 et \hat{x}_2 capture une partie de l'argument pour produire la substitution $\{x_1 := \hat{c}, x_2 := (\langle z \rangle \hat{z}.z \hat{c})\}$.

Enfin, le résultat du filtrage d'un argument a par un motif p est obtenu après un *test* sur le filtrage composé qui vérifie que toutes les variables de filtrage ont été pourvues, c'est-à-dire que le motif p contient bien toutes les *matchables* dont le nom est lié par θ . Nous verrons à l'exemple 6.19 que des problèmes peuvent sinon survenir lors de la réduction.

Définition 6.16 (Filtrage). *Le **filtrage** d'un argument a par un motif p avec variables de filtrage θ , noté $\{a/\theta p\}$, est défini par les cas suivants :*

- Si $\{\{a/\theta p\}\} = \perp$, alors $\{a/\theta p\} = \perp$.
- Si $\{\{a/\theta p\}\} = \sigma$ et $\theta = \text{dom}(\sigma)$, alors $\{a/\theta p\} = \sigma$.
- Si $\{\{a/\theta p\}\} = \sigma$ et $\theta \neq \text{dom}(\sigma)$, alors $\{a/\theta p\} = \perp$.

Remarquons que le filtrage de a par p n'est pas défini si $\{\{a/\theta p\}\} = \Sigma$.

Comme en λ -calcul, la relation de réduction est définie ici par une seule méta-opération. Elle est appelée β_m et réalise en une étape le filtrage et la substitution.

Définition 6.17 (Réduction). *Tout sous-terme de la forme $r = (\langle\theta\rangle p.b)a$ est appelé un **pré-radical**. Si $\{a/\theta p\}$ est défini alors le pré-radical r est un **radical** et la règle β_m s'applique. Pour tout terme b on définit b^\perp comme étant égal à une certaine forme normale close fixée t_\perp .*

$$r = (\langle\theta\rangle p.b)a \xrightarrow{\beta_m} b^{\{a/\theta p\}}$$

La relation de réduction dans n'importe quel contexte est formalisée par les règles d'inférence suivantes :

$$\frac{t_1 \xrightarrow{\beta} t'_1}{t_1 t_2 \xrightarrow{\beta} t'_1 t_2} (\mu) \quad \frac{t_2 \xrightarrow{\beta} t'_2}{t_1 t_2 \xrightarrow{\beta} t_1 t'_2} (\nu)$$

$$\frac{p \xrightarrow{\beta} p'}{\langle\theta\rangle p.b \xrightarrow{\beta} \langle\theta\rangle p'.b} (\zeta)$$

$$\frac{b \xrightarrow{\beta} b'}{\langle\theta\rangle p.b \xrightarrow{\beta} \langle\theta\rangle p.b'} (\xi)$$

Par convention le terme d'erreur est l'identité : $t_\perp = I = \langle x \rangle \hat{x}.x$. Ceci permet en particulier de rattraper les erreurs de filtrage pour déclencher des cas alternatifs ou par défaut [JK09].

Exemple 6.18 (Exemple filé).

Les fragments de l'exemple 6.15 sont rassemblés ici. Les radicaux contractés sont soulignés.

$$\frac{(\langle x_1 x_2 \rangle \hat{x}_1 \hat{x}_2.(\langle y \rangle x_2 \hat{y}.b)\hat{c}) (\hat{c}(\langle z \rangle \hat{z}.z\hat{c}))}{\rightarrow_{\beta_m} (\langle y \rangle (\langle z \rangle \hat{z}.z\hat{c})\hat{y}.b)\hat{c}} \quad (\text{RE1})$$

$$\rightarrow_{\beta_m} (\langle y \rangle \hat{y}\hat{c}.b)\hat{c} \quad (\text{RE2})$$

$$\rightarrow_{\beta_m} (\langle y \rangle \hat{y}\hat{c}.b)\hat{c} \quad (\text{RE3})$$

$$\rightarrow_{\beta_m} \perp \quad (\text{RE4})$$

L'exemple 6.19 suivant montre l'enjeu du test effectué pour déduire le résultat d'un filtrage $\{a/\theta p\}$ à partir du résultat du filtrage composé $\{\{a/\theta p\}\}$ (définition 6.16).

Exemple 6.19.

Considérons le terme suivant :

$$t = (\langle x, y \rangle \hat{x}.xy)\hat{c}$$

Le terme t est un pré-radical qui déclenche le filtrage $\{\hat{c}/\theta \hat{x}\}$ de l'argument \hat{c} par le motif \hat{x} avec les variables de filtrage x et y . Le filtrage composé est défini et réussit :

$$\{\{\hat{c}/\theta \hat{x}\}\} = \{x := \hat{c}\}$$

En revanche, le domaine de la substitution $\sigma = \{x := \hat{c}\}$ obtenue ne contient pas y et nous avons donc :

$$\{\hat{c}/\theta \hat{x}\} = \perp$$

Si nous passions outre ce test et essayions d'appliquer la substitution σ au corps de la fonction nous obtiendrions

$$t' = (xy)^\sigma = \hat{c}y$$

Le pas de réduction de source t et de cible t' ferait alors apparaître une variable libre y à partir de matériel clos.

Remarque 6.20 ([JK09]). Le λ -calcul s'encode directement dans PPC : tout λ -terme de la forme $\lambda x.t$ peut être écrit comme un PPC-terme $\langle x \rangle \hat{x}.t$. Pour retrouver la β -réduction il suffit de remarquer que l'application $(\langle x \rangle \hat{x}.t)u$ génère le filtrage trivial $\{u/x \hat{x}\}$ qui se résout toujours en la substitution $\{x := u\}$.

La notion habituelle de descendance s'adapte aux constructions de PPC.

Définition 6.21 (Descendance). Soit $\rho : t \xrightarrow{r} t'$ avec $r = (\langle \theta \rangle p.b)a \rightarrow_{\beta_m} r'$, et q_r la position de r dans t . Soit q un position dans t .

- Si q est disjointe de q_r , alors $q/\rho = \{q\}$.
- Si q est un préfixe strict de q_r , alors $q/\rho = \{q\}$.
- Si $\{a/\theta p\} = \sigma$ et q est de la forme $q_r \cdot 12 \cdot q'$ et tel que $b|_q$ n'est pas une variables, alors $q/\rho = \{q_r \cdot q'\}$.
- Si $\{a/\theta p\} = \sigma$ et q est de la forme $q_r \cdot 2 \cdot q_{\hat{x}} \cdot q'$ et tel que $p|_{q_{\hat{x}}}$ est une matchable \hat{x} liée par θ , alors soit Q_x l'ensemble des positions des occurrences libres de x dans t , $q/\rho = q_r \cdot Q_x \cdot q'$.
- Dans tous les autres cas $q/\rho = \emptyset$

Comme d'habitude, en parlant de descendance nous identifions une position avec le sous-terme à cette position.

Exemple 6.22.

Dans l'exemple filé 6.18, le sous-terme $(\langle z \rangle \hat{z}.z\hat{c})\hat{y}$ dans (RE2) est l'unique descendant du sous-terme $x_2\hat{y}$ de (RE1). Il s'agit également d'un radical créé par ce pas de réduction, et non un résidu, car l'ancêtre $x_2\hat{y}$ n'est pas un radical. À l'inverse, l'application $\hat{c}(\langle z \rangle \hat{z}.z\hat{c})$ de (RE1) est détruite par le filtrage et n'a pas de descendant.

La définition de la réduction faible pour le *Pure Pattern Calculus* demande une rapide discussion. En effet, deux règles de contexte de *PPC* permettent de passer *sous un lieu*. La règle (ζ) permet la réduction d'un motif et la règle (ξ) permet la réduction du corps d'une fonction. Dans la version faible du *Pure Pattern Calculus* nous n'empêcherons en rien la réduction dans un motif par (ζ). En effet, il faut remarquer d'une part que par le principe même des motifs dynamiques cette réduction peut être nécessaire à l'application d'une fonction, et d'autre part que les occurrences liées dans le motif sont des *matchables* qui ne peuvent pas être substituées et non des variables ordinaires. Ainsi, la seule règle (ξ) est contrainte d'une manière inspirée par la réduction faible combinatoire du λ -calcul [cH98].

Définition 6.23 (Réduction faible). *La réduction faible du Pure Pattern Calculus est la relation de réduction obtenue en remplaçant la règle d'inférence (ξ) par la règle contrainte (ξ') :*

$$\frac{b \xrightarrow{r} b' \quad \theta \cap \text{fv}(r) = \emptyset}{\langle \theta \rangle p.b \xrightarrow{r} \langle \theta \rangle p.b'} (\xi')$$

Dans ce cadre, un pré-radical $r = (\langle \theta \rangle p.b)a$ dans un terme t est un radical si à la fois $\{a/\theta p\}$ est défini et r est libre dans t . Remarquons que cette condition de liberté concerne uniquement les variables et non les *matchables*. C'est en effet bien uniquement sur les variables que seront définis les squelettes et les positions gelées.

6.3 Causalité dans le *Pure Pattern Calculus*

Dans le cas simple du λ -calcul, un pré-radical apparaît dès qu'une abstraction est appliquée à un argument. Comme nous l'avons vu au chapitre 5, dans ce cas la partie de l'histoire qui est pertinente pour la création de ce pré-radical peut être enregistrée localement, ainsi que dans les λ -calculs étiquetés de Lévy [Lé78] et Blanc, Lévy et Maranget [BLM07]. Cependant, dans le cas du filtrage un argument voué à être comparé à un certain motif doit avoir une certaine forme, qui dépend dudit motif. Ceci est illustré par l'exemple 6.24.

Exemple 6.24.

Considérons le programme suivant :

```
let id x = x
and rec count = fun
  | Data a   -> 1
  | Node l r -> (count l) + (count r)
in
  count (id (Data 0))
```

La fonction *count* est appelée avec l'argument *id (Data 0)*, qui doit être évalué à *Data 0* avant que l'application ne puisse être résolue.

Ainsi, dans ce cadre l'histoire pertinente d'un radical peut également dépendre de l'histoire de ses sous-termes, et ce jusqu'à une profondeur arbitraire. Remarquons de plus que l'exemple 6.24 ne repose sur aucune des particularités de polymorphisme de PPC : tout ceci est déjà valable pour les mécanismes de filtrage usuels.

Nous devons donc revoir l'analyse des créations de radicaux. Commençons par constater que les cas (1) de production d'une abstraction à gauche d'une application et (2⁺) de substitution affectant une application qui ont été énoncés au chapitre 5 pour le λ -calcul faible (exemples 5.2 et 5.3 et lemme 5.4 de création de radicaux) sont toujours valables dans le cas du *Pure Pattern Calculus* faible.

Exemple 6.25 (Transcription libre de l'exemple 5.2). _____

Le cas de création (1) d'un pré-radical de PPC couvre toujours l'équivalent des deux premiers cas de Lévy :

$$\begin{aligned} ((\langle \emptyset \rangle \hat{c}.(\langle \theta \rangle p.b)\hat{c})\underline{a}) &\rightarrow (\langle \theta \rangle p.b)a \\ ((\langle x \rangle \hat{x}.x)(\langle \theta \rangle p.b)\underline{a}) &\rightarrow (\langle \theta \rangle p.b)a \end{aligned}$$

Exemple 6.26 (Transcription libre de l'exemple 5.3). _____

Cas (2⁺), vu dans PPC : un pré-radical $(\langle x \rangle \hat{x}.yx)\underline{a}$ n'est pas un radical pour la seule raison de la présence d'une occurrence libre d'une variable y liée au dessus.

$$(\langle y \rangle \hat{y}.((\langle x \rangle \hat{x}.yx)\underline{a}))t \rightarrow (\langle x \rangle \hat{x}.tx)a$$

Ceci contient toujours le troisième cas de Lévy :

$$(\langle y \rangle \hat{y}.(y\underline{a}))(\langle \theta \rangle p.b) \rightarrow (\langle \theta \rangle p.b)a$$

Nous pouvons également observer que ces deux premiers cas de créations (1) et (2⁺) couvrent déjà certains traits caractéristiques de PPC et du polymorphisme de motif.

Exemple 6.27. _____

Une erreur de filtrage peut créer un nouveau radical par le cas (1), qui peut être utilisé pour déclencher un cas alternatif lors d'une définition à plusieurs cas [JK09].

$$(((\langle \emptyset \rangle \hat{c}.(\langle \theta \rangle p.b)\hat{c})\underline{a}) \rightarrow t_{\perp} a = (\langle x \rangle \hat{x}.x)a$$

L'instanciation d'un motif paramétré avec un constructeur peut créer un radical selon le cas (2⁺) :

$$(\langle y \rangle \hat{y}.((\langle x \rangle y\hat{x}.b)\hat{c}\underline{a}))\hat{c} \rightarrow (\langle x \rangle \hat{x}.b)\hat{c}a$$

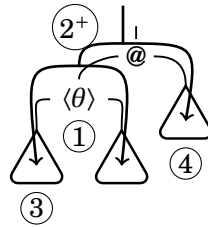
Au-delà de ces cas (1) et (2⁺), le filtrage d'une part et les motifs dynamiques d'autre part apportent chacun un nouveau cas de création, et ces deux cas sont symétriques. En effet, quand l'abstraction d'un certain motif p est appliquée à un argument a , alors a doit être comparé à p . Cependant, comme nous l'avons vu à la section 6.2, le filtrage $\{a/\theta p\}$ n'est pas toujours défini. Il se peut que certains sous-termes de a ou de p doivent être en forme testable, ce qui peut requérir une réduction plus avancée de a ou de p . Dans ce cas, le filtrage $\{a/\theta p\}$ ne sera défini qu'après certaines réductions de a ou de p , ce qui signifie que le radical correspondant à ce filtrage pourra être créé par des réductions ayant lieu à l'intérieur du motif ou de l'argument.

Les deux nouveaux cas de création sont donc : (3) la réduction dans le motif d'un pré-radical, et (4) la réduction dans l'argument d'un pré-radical. Le seul autre cas de forme non testable est en effet la présence d'une occurrence de variable qui doit être substituée, et nous avons vu à l'exemple 6.27 comment ce cas était couvert par le cas (2⁺).

Avant d'énoncer formellement au lemme 6.29 ces différents cas de création de radicaux dans PPC, visualisons-les sur un schéma. Dans les schémas suivants, une abstraction $\langle\theta\rangle p.b$ sera dénotée par un nœud binaire $\langle\theta\rangle$ dont le fils gauche représente le motif p et le fils droit le corps b , comme ci-dessous :

$$\langle\theta\rangle p.b \equiv \begin{array}{c} \langle\theta\rangle \\ \swarrow \quad \searrow \\ \triangle p \quad \triangle b \end{array}$$

Les quatre cas de création évoqués peuvent être localisés comme suit sur le schéma d'un radical :



Exemple 6.28.

Dans l'exemplé filé 6.18, le radical du terme (RE2) est créé par le premier pas de réduction selon le cas (2⁺), et le radical du terme (RE3) est créé par le deuxième pas de réduction selon le cas (3).

Remarquons ici que le cas (3) est spécifique aux motifs dynamiques, tandis que le cas (4) existe déjà avec les mécanismes de filtrage usuels utilisant des motifs algébriques, ainsi que nous l'avons vu dans l'exemple 6.24.

Le lemme suivant sanctionne le description des créations de radicaux que nous venons de décrire.

Lemme 6.29 (Création de radicaux). Soit $\rho : t = c[r] \rightarrow c[r'] = t'$ un pas de réduction dans PPC avec $r = (\langle\theta\rangle p.b)a$ et $r' = b^\mu$. Soit $r_c = (\langle\theta_c\rangle p_c.b_c)a_c$ un radical de t' créé par ρ . Notons $t' = c_c[r_c]$. Alors l'une des propositions suivantes est vérifiée.

- (1) $c = c_c[\Box a_c]$ et $b^\mu = \langle \theta_c \rangle p_c . b_c$.
- (2⁺) La position de r_c descend d'une position de la θ -épine de b .
- (3) Il existe un contexte c' tel que $c = c_c[\langle \theta_c \rangle c' . b_c] a_c$ et $\{a_c /_\theta c'[\langle \theta \rangle p . b] a\}$ n'est pas défini.
- (4) Il existe un contexte c' tel que $c = c_c[\langle \theta_c \rangle p_c . b_c] c'$ et $\{c'[\langle \theta \rangle p . b] a /_\theta p_c\}$ n'est pas défini.

Pour la preuve, commençons comme à la section 5.2 par isoler le cas de la création d'un radical à la racine du terme.

Lemme 6.30. *Sous les hypothèses du lemme 6.29, si de plus $c_c = \Box$ et $c \neq \Box$ alors l'une des propositions (1), (3), ou (4) est vérifiée.*

Démonstration. Remarquons d'abord que $c_c = \Box$ implique $r_c = t' = c[b^\mu]$. Raisonnons par cas sur $c \neq \Box$.

- Cas $c = \langle \theta_0 \rangle c' . b_0$ ou $c = \langle \theta_0 \rangle p_0 . c'$. Alors il existe une paire p'_0, b'_0 telle que $t' = \langle \theta_0 \rangle p'_0 . b'_0$, et en particulier t' n'est pas un pré-radical. Ceci n'est pas notre cas.
- Cas $c = \langle \theta_c \rangle p_c . b_c] c'$. Comme t n'est pas un radical (car r_c est créé par ρ), $\{c'[\langle \theta \rangle p . b] a /_\theta p_c\}$ n'est pas défini et la proposition (4) est vérifiée.
- Cas $c = c' a_c$. Raisonnons par cas sur c' .
 - Cas $c' = \Box$, c'est-à-dire $c = \Box a_c$. Alors $b^\mu = \langle \theta_c \rangle p_c . b_c$ et la proposition (1) est vérifiée.
 - Cas $c' = t_1 c''$ ou $c' = c'' t_2$, c'est-à-dire $c = t_1 c'' t_0$ ou $c = c'' t_2 t_0$. Alors il existe une paire t'_1, t'_2 telle que $c[b^\mu] = t'_1 t'_2 t_0$. Ceci n'est pas notre cas.
 - Cas $c' = \langle \theta_c \rangle p_c . c''$, c'est-à-dire $c = \langle \theta_c \rangle p_c . c''] a_c$. Comme le pré-radical t n'est pas un radical, $\{a_c /_\theta p_c\}$ n'est pas défini. Cela ne change pas dans $t' = \langle \theta_c \rangle p_c . c''[b^\mu] a_c$, donc t' n'est pas un radical. Ceci n'est pas notre cas.
 - Cas $c' = \langle \theta_c \rangle c'' . b_c$, c'est-à-dire $c = \langle \theta_c \rangle c'' . b_c] a_c$. Comme le pré-radical t n'est pas un radical, $\{a_c /_\theta c''[\langle \theta \rangle p . b] a\}$ n'est pas défini et la proposition (3) est vérifiée. \square

Démonstration du lemme 6.29. Raisonnons par cas sur les positions relatives de r' et r_c .

- Si r' et r_c sont disjoints, alors le radical r_c existe déjà dans t . Ceci n'est pas notre cas.
- Si r' est un sous-terme de r_c , alors il existe c' tel que $r_c = c'[r']$ et $t' = c_c[c'[r']]$. Comme r_c est créé par ρ , son ancêtre $c'[r]$ n'est pas un radical dans t . Supposons que $c'[r]$ soit un pré-radical gelé. Alors il existe dans $c'[r]$ une occurrence d'une variable x liée par un lieu de c_c . Raisonnons par cas sur la position de cette variable.
 - Si x apparaît dans r , alors le pré-radical r est lui-même gelé, ce qui n'est pas notre cas.
 - Si x apparaît dans c' , alors le pré-radical $c'[r']$ contient toujours x et est donc toujours gelé, ce qui n'est pas notre cas.

Donc $c'[r]$ n'est pas un pré-radical gelé, et nous pouvons appliquer le lemme 6.30 au pas de réduction $c'[r] \rightarrow c'[r']$.

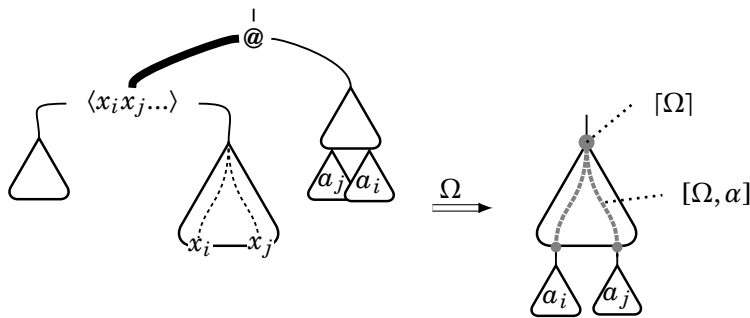
- Si r_c est un sous-terme de $r' = b^\mu$, alors en remarquant que le filtrat μ est une substitution σ il reste trois cas à considérer.
 - Si r_c est dans le codomaine de σ , alors son ancêtre est déjà un radical. Ceci n'est pas notre cas.
 - Si la position de r_c est dans la $\{\theta\}$ -épine de b , alors la proposition (2⁺) est vérifiée.
 - Si la position de r_c est dans b mais n'est pas dans la $\{\theta\}$ -épine de u , alors l'ancêtre de r_c lui est égal et est déjà un radical. Ceci n'est pas notre cas. □

Comme cela a été fait au chapitre 5 pour le λ -calcul faible, un étiquetage du *Pure Pattern Calculus* peut être construit par inversion de la description ci-dessus. La question n'est plus alors *d'où ce radical vient-il ?* mais *quelles peuvent être les conséquences futures de cette étape de réduction ?* Si chaque radical réduit laisse son empreinte à tout endroit où il peut contribuer à quelque chose (et à aucun autre endroit !), alors l'origine d'un radical peut être reconstituée par une collecte des informations qui ont été laissées à cette place par les pas de réduction passés.

Pour réaliser ceci nous allons ajouter des étiquettes à la syntaxe de *PPC*, de manière que chaque étiquette enregistre les informations localement pertinentes à propos des réductions passées, et que les réductions agissent sur ces étiquettes pour maintenir les informations à jour.

Comme au chapitre 5 mais à la différence des chapitres 2 à 4 précédents, un *nom* est attribué à chaque radical. Ce nom est déduit des étiquettes du radical et sert à marquer les potentielles contributions futures.

La réduction d'un radical r de nom Ω transforme les étiquettes du réduit. D'abord r peut contribuer à quelque chose à sa racine, par le cas (1). Une étiquette $[\Omega]$ dénotant l'*épicentre* de Ω est ajoutée à la racine du réduit pour témoigner de ceci. De plus, si r est un filtrage réussi, alors il produit une substitution. Dans ce cas r peut contribuer à quelque chose par le cas (2⁺), à n'importe quelle position le long du chemin de propagation de la substitution, qui correspond à une épine dans le vocabulaire des chapitres précédents. Cet autre type de contribution sera marqué dans les étiquettes par la construction $[\Omega, \alpha]$ que nous avons déjà vue, qui dénote rappelons-le une *copie de α déclenchée par Ω* . Le schéma suivant résume ceci :



Pour l'instant, nous n'avons qu'une transcription directe de l'étiquetage construit au chapitre 5. Les contributions des réductions du motif et de l'argument ne sont pas encore visibles ici. En effet, ces contributions ne concernent pas le réduit du radical mais d'autres positions indéterminées du contexte.

La conception d'un mécanisme tenant compte de ces contributions non locales est l'une des difficultés importantes de ce chapitre, difficulté qui est due notamment au fait que *PPC* n'est pas défini par une règle *CRS* usuelle mais par une méta-opération complexe. Cette difficulté pourrait être contournée en passant par un encodage de *PPC* dans un cadre plus explicite, notamment au moyen de la traduction dans les *HRS* proposée par van Oostrom et van Raamsdonk [vOvR11]. Nous ne nous autoriserons cependant pas cette simplification ici, et profiterons plutôt de cette occasion pour tester l'expressivité du cadre axiomatique des *SLS*.

Ainsi, aux étiquettes laissées dans les termes à l'attention des pas de réduction futurs, nous ajoutons un mécanisme de collecte des étiquettes laissées par les pas de réduction passés dans le motif et dans l'argument : toute information pertinente trouvée dans son motif p ou dans son argument a sera incluse dans le nom Ω d'un radical $(\langle\theta\rangle p.b)a$. La difficulté que nous devons résoudre alors est de discriminer dans le motif p et l'argument a les informations pertinentes des informations non pertinentes. En particulier, ces informations ne seront pas les mêmes en cas de réussite ou d'échec du filtrage, ce qui sera illustré par les exemples 6.34 à 6.36. Nous verrons notamment que les préfixes pertinents du motif p et de l'argument a sont mutuellement dépendants, et sont déterminés dynamiquement par la méta-opération de filtrage elle-même. C'est donc à cette méta-opération que reviendra la tâche de collecter les informations pertinentes contenues dans le motif p et l'argument a . Les autres aspects de la définition du système ayant déjà été essentiellement explorés au chapitre 5 précédent, c'est cet enrichissement de l'opération de filtrage composé qui est le premier enjeu de la section 6.4 suivante.

6.4 Filtrage étiqueté

Cette section introduit la syntaxe du *Pure Pattern Calculus* étiqueté, et définit une opération de filtrage composé qui collecte à la volée les étiquettes pertinentes du motif et de l'argument.

La syntaxe de *PPC* est augmentée des étiquettes introduites à la section précédente, qui sont identiques à celles employées au chapitre 5. De même, nous n'allons pas ici faire porter les étiquettes par les symboles de la syntaxe d'origine, mais plutôt leur donner des positions propres, et distinguer dans la grammaire deux catégories syntaxiques à part caractérisant les termes étiquetés qui ont ou n'ont pas une étiquette à leur racine.

Définition 6.31 (Étiquettes et termes étiquetés).

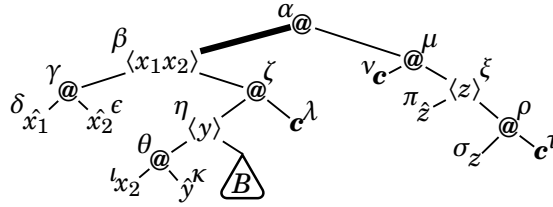
$\alpha ::= p \mid [\Omega] \mid [\Omega, \alpha]$	Étiquettes
$\Gamma, \Delta, \Omega ::= \alpha_1 \alpha_2 \dots \alpha_n$	Séquences d'étiquettes
$X, Y, Z ::= T \mid N$	Termes étiquetés
$N ::= \hat{x} \mid TT \mid \langle \theta \rangle T.T$	Termes étiquetés incomplets
$A, B, P, R, T ::= x \mid \alpha : X$	Termes étiquetés complets

où n est un entiers strictement positif, $x, y, z \in \mathcal{X}$ sont des variables, θ, τ sont des séquences de variables, et p est une position. Comme avant, un **terme initial** est un terme dont les étiquettes sont toutes initiales et différentes. Pour toute séquence d'étiquettes $\Gamma = \gamma_1 \dots \gamma_n$ et tout terme X , nous abrègerons $\gamma_1 : \dots : \gamma_n : X$ en $\Gamma \cdot X$.

Les lettres P, B , et A sont toujours utilisées pour dénoter des termes ayant un rôle de motif, de corps de fonction ou d'argument, et la lettre grecque Ω indique le nom d'un radical. Les étiquettes peuvent ici être vues comme des membres à part entière de la signature du système, et les notions de positions sur des termes syntaxiques les prennent donc en compte. Les notions de variables et *matchables* libres, d' α -conversion et de substitution sont héritées naturellement de PPC (définitions 6.9, 6.10, et 6.12).

Exemple 6.32.

Ce schéma montre un étiquetage que nous pouvons supposer initial du terme (RE1) de l'exemple filé 6.18. Ce terme étiqueté sera nommé (L1) dans les exemples subséquents.



Notons que ce schéma est compressé : en réalité les étiquettes sont des nœuds à part entière intercalés entre les nœuds principaux.

Comme au chapitre 5 (définition 5.8), la relation de *contribution directe* \hookrightarrow^d est une relation sur les séquences d'étiquettes. Rappelons sa définition.

Définition 6.33 (Contribution). Une séquence d'étiquettes Ω **contribue directement** à une séquence d'étiquettes Γ , noté $\Omega \hookrightarrow^d \Gamma$, si et seulement s'il existe deux séquences d'étiquettes éventuellement vides Γ_1 et Γ_2 telles que $\Gamma = \Gamma_1[\Omega]\Gamma_2$ ou $\Gamma = \Gamma_1[\Omega, \alpha]\Gamma_2$. Nous notons \hookrightarrow et appelons relation de **contribution** la clôture transitive de \hookrightarrow^d .

Il nous faut maintenant définir une notion de filtrage composé étiqueté sur les termes étiquetés, qui d'une part effectue sur les termes le même travail que le filtrage composé de PPC, et d'autre part collecte les étiquettes utiles du motif et de l'argument. Un filtrage composé étiqueté $\llbracket A/\theta P \rrbracket$, s'il ne renvoie pas \bar{X} , va renvoyer une paire (Δ, μ) formée d'une séquence d'étiquettes Δ et d'un

filtrat défini μ . La séquence Δ contient les informations de contribution relatives aux réductions du motif et de l'argument, que la méta-opération de filtrage a collectées à la volée.

La discipline proposée pour faire intervenir les étiquettes du préfixe et de l'argument d'un radical dans son nom est la suivante : en cas de filtrage réussi, c'est-à-dire produisant une substitution, toutes les étiquettes des positions explorées par l'opération de filtrage composé doivent être retenues dans le nom. En cas d'échec de filtrage en revanche, c'est-à-dire d'un filtrage renvoyant \perp , ne seront conservées que les étiquettes des positions qu'il est nécessaire d'explorer pour atteindre chacune des erreurs présentes. Dans ce dernier cas nous ignorerons donc en particulier les étiquettes des branches du filtrage qui ne participent pas à son échec. Enfin, il n'y a pas de radical si le filtrage renvoie Σ , et la question du nom ne se pose alors pas. Nous discuterons à la section 6.7 certaines variantes possibles.

Dans tous les cas, la clé est de collecter les étiquettes des positions « visitées » par le filtrage composé étiqueté. Cependant, cette formulation contient un piège, car si certaines positions sont explicitement visitées par le filtrage, d'autres ne le sont qu'implicitement. En effet, plusieurs équations de la définition du filtrage composé invoquent des tests de testabilités du motif et/ou de l'argument, tests qui impliquent une inspection implicite d'une partie de la structure des termes concernés, comme le montrent les exemples 6.34 à 6.36.

Exemple 6.34.

Considérons le filtrage

$$\{\hat{c}(\hat{c}X)/_x \hat{c}\hat{x}\}$$

et sa version étiquetée

$$\{\alpha : @(\beta : \hat{c}, \gamma : @(\delta : \hat{c}, \zeta : X))/_x \eta : @(\kappa : \hat{c}, \mu : \hat{x})\}$$

où X est un terme étiqueté quelconque. Nous attendons de ce filtrage qu'il réussisse et génère la substitution $\{x := \gamma : @(\delta : \hat{c}, \zeta : X)\}$. Décomposons la méta-opération de filtrage composé (sans les étiquettes) pour observer les positions visitées par le filtrage :

$$\begin{aligned} & \{\hat{c}(\hat{c}X)/_x \hat{c}\hat{x}\} \\ = & \{\hat{c}/_x \hat{c}\} \uplus \{\hat{c}X/_x \hat{x}\} && \text{car } \hat{c}(\hat{c}X) \text{ et } \hat{c}\hat{x} \text{ testables} \\ = & \{\} \uplus \{\hat{c}X/_x \hat{x}\} && \text{car } \hat{c} \notin \{x\} \\ = & \{x := \hat{c}X\} && \text{car } x \in \{x\} \end{aligned}$$

Nous avons d'abord une décomposition de l'application externe, puis une comparaison des deux symboles \hat{c} , et enfin une reconnaissance de la variable de filtrage \hat{x} . Notons que la position de l'application ($\hat{c}X$) n'est pas examinée. Ainsi, nous devons collecter ici toutes les étiquettes sauf γ , δ , et ζ .

Exemple 6.35.

Considérons une variation du filtrage de l'exemple précédent où les applications ont été retournées.

$$\{(\hat{c}X)\hat{c}/_x \hat{c}\hat{x}\}$$

$$\{\alpha : @(\gamma : @(\delta : \hat{c}, \zeta : X), \beta : \hat{c}) /_x \eta : @(\mu : \hat{x}, \kappa : \hat{c})\}$$

Nous attendons de ce filtrage qu'il réussisse encore et génère la même substitution $\{x := \beta : @(\gamma : \hat{c}, \delta : X)\}$. La méta-opération de filtrage se déroule alors de la même manière, mais avec un point à noter : la vérification de testabilité de l'argument se fait sur $(\hat{c}X)\hat{c}$ plutôt que sur $\hat{c}(\hat{c}X)$. Par conséquent, la structure du sous-argument $\hat{c}X$ doit être examinée pour découvrir le constructeur \hat{c} le plus à gauche, ce qui signifie que les étiquettes γ et δ sont maintenant visitées implicitement et doivent être collectées. Seule l'étiquette ζ peut finalement être omise.

Exemple 6.36.

Considérons enfin le filtrage

$$\{(\hat{c}\hat{c})\hat{c} /_{\emptyset} (\hat{c}\hat{c})\hat{d}\}$$

$$\{\alpha : @(\beta : @(\gamma : \hat{c}, \delta : \hat{c}), \zeta : \hat{c}) /_{\emptyset} \eta : @(\kappa : @(\mu : \hat{c}, \nu : \hat{c}), \xi : \hat{d})\}$$

Il échoue à cause de la branche la plus à droite $\{\delta : \hat{c} /_{\theta} \xi : \hat{d}\}$. Cependant, la découverte de cette erreur nécessite la décomposition de l'application externe, qui elle-même dépend de la testabilité de l'ensemble. Ainsi la branche la plus à gauche doit implicitement être également intégralement explorée, et seules les positions des étiquettes δ et ν peuvent être ignorées.

La manière la plus simple d'intégrer au filtrage composé une collecte de toutes les étiquettes pertinentes du motif consiste donc à accumuler d'une part les étiquettes des positions explicitement visitées par le filtrage composé étiqueté, et d'autre part les étiquettes des positions implicitement visitées par des tests de testabilité, ces dernières formant des séquences d'étiquettes que nous appellerons *témoins de testabilité* (définition 6.38 à venir).

Pour formaliser ceci, étendons à la syntaxe étiquetée les notions de structures de données et de formes testables.

Définition 6.37 (Formes testables étiquetées). *Les formes testables étiquetées sont définies par la grammaire suivante.*

$$\begin{array}{ll} D & ::= \hat{x} \mid D_1 T \mid D_l \quad \text{Structures de données étiquetées} \\ D_l & ::= \alpha : D \quad \text{Structures de données complètement étiquetées} \\ M & ::= D \mid \langle \theta \rangle P.B \mid \alpha : M \quad \text{Formes testables étiquetées} \end{array}$$

À chaque forme testable étiquetée M est associée une séquence d'étiquette appelée *témoin de testabilité*, dont le but est de collecter toutes les informations concernant les événements passés qui ont contribué à mettre M sous forme testable.

Définition 6.38 (Témoin de testabilité). *Le **témoin de testabilité** d'une forme testable étiquetée M , noté $|M|$, est défini par les équations suivantes.*

$$\begin{array}{ll} |\alpha : Z| & = \alpha |Z| \\ |\hat{x}| & = \varepsilon \\ |T_1 T_2| & = |T_1| \\ |\langle \theta \rangle P.B| & = \varepsilon \end{array}$$

L'idée directrice de la collecte effectuée à la volée par le filtrage composé étiqueté est donc : accumuler les étiquettes des positions explicitement visitées, et les témoins de testabilité des sous-termes dont la testabilité est demandée. Nous pouvons voir à l'exemple 6.39 que cette technique est fortement redondante.

Exemple 6.39.

Reprenons le filtrage de l'exemple 6.36 précédent

$$\{\alpha : @(\beta : @(\gamma : \hat{c}, \delta : \hat{c}), \zeta : \hat{c}) /_{\theta} \eta : @(\kappa : @(\mu : \hat{c}, \nu : \hat{c}), \xi : \hat{d})\}$$

Sa première étape décompose l'application externe après vérification des testabilités du motif $\eta : @(\kappa : @(\mu : \hat{c}, \nu : \hat{c}), \xi : \hat{d})$ et de l'argument $\alpha : @(\beta : @(\gamma : \hat{c}, \delta : \hat{c}), \zeta : \hat{c})$, ce qui incite à collecter les témoins de testabilité $\eta\kappa\mu$ et $\alpha\beta\gamma$. La deuxième étape, de décomposition encore, vérifie la testabilité du sous-motif $\kappa : @(\mu : \hat{c}, \nu : \hat{c})$ et du sous-argument $\beta : @(\gamma : \hat{c}, \delta : \hat{c})$, incitant à la collecte des témoins de testabilité $\kappa\mu$ et $\beta\gamma$. Ces deux derniers témoins de testabilité sont strictement recouverts par les deux premiers. En effet, le premier test de testabilité contenait le second.

Nous pourrions botter en touche et présenter Δ non comme une séquence mais comme un ensemble d'étiquettes collectées. La solution est tout à fait légitime, et ne demande qu'une complexification mineure de la structure des étiquettes. Ceci dit, traiter le problème de front est motif à explorer un peu plus la structure du *Pure Pattern Calculus*. Accordons-nous cet instant de coquetterie.

Les redondances observées ont deux causes :

- Comme vu à l'exemple 6.39, nous faisons certains tests de testabilité plusieurs fois. Ce phénomène est lié à la définition du filtrage composé du *Pure Pattern Calculus*.
- Une collecte naïve tendrait à enregistrer plusieurs fois une étiquette qui serait d'abord visitée implicitement par un test de testabilité puis visitée explicitement par le filtrage composé.

Les redondances des tests de testabilité pourraient être annulées en se basant sur une définition alternative du filtrage composé de *PPC*, dans laquelle les structures de données ne seraient pas décomposées argument par argument comme le suggère l'équation

$$\{\{a_1 a_2 /_{\theta} p_1 p_2\}\} = \{\{a_1 /_{\theta} p_1\}\} \uplus \{\{a_2 /_{\theta} p_2\}\}$$

mais seraient décomposées en une seule fois, par une équation de la forme

$$\{\{a_1 \dots a_n /_{\theta} p_1 \dots p_n\}\} = \{\{a_1 /_{\theta} p_1\}\} \uplus \dots \uplus \{\{a_n /_{\theta} p_n\}\}$$

Ainsi, le test de testabilité qui est associé à l'une comme à l'autre de ces deux équations pourrait être factorisé dans le deuxième cas.

Exemple 6.40.

Le filtrage composé de l'exemple 6.39

$$\begin{aligned} & \{(\hat{c}\hat{c})\hat{c} /_{\theta} (\hat{c}\hat{c})\hat{d}\} \\ = & \{\{\hat{c}\hat{c} /_{\theta} \hat{c}\hat{c}\}\} \uplus \{\{\hat{c} /_{\theta} \hat{d}\}\} && \text{car } \hat{c}\hat{c} \text{ et } \hat{c}\hat{c}\hat{d} \text{ testables} \\ = & \{\{\hat{c} /_{\theta} \hat{c}\}\} \uplus \{\{\hat{c} /_{\theta} \hat{c}\}\} \uplus \{\{\hat{c} /_{\theta} \hat{d}\}\} && \text{car } \hat{c}\hat{c} \text{ et } \hat{c}\hat{c} \text{ testables} \end{aligned}$$

pourrait être raccourci en

$$\begin{aligned} & \{(\hat{c}\hat{c})\hat{c}/_{\theta}(\hat{c}\hat{c})\hat{d}\} \\ = & \{ \hat{c}/_{\theta} \hat{c} \} \uplus \{ \hat{c}/_{\theta} \hat{c} \} \uplus \{ \hat{c}/_{\theta} \hat{d} \} \quad \text{car } \hat{c}\hat{c} \text{ et } \hat{c}\hat{c}\hat{d} \text{ testables} \end{aligned}$$

Et dans ce cas la testabilité des sous-termes $\kappa : @(\mu : \hat{c}, \nu : \hat{c})$ et $\beta : @(\gamma : \hat{c}, \delta : \hat{c})$ n'aurait pas besoin d'être vérifiée.

Nous verrons cependant que cette transformation de la définition n'est pas nécessaire ici. Nous pouvons en effet éliminer d'un coup les deux sources de redondances, en déclinant le filtrage composé étiqueté $\{Y/\theta X\}$ en deux mode : un premier à appliquer par défaut, noté $\{Y/\theta X\}_d$, et qui collecte toutes les étiquettes qu'il visite et tous les témoins des tests de testabilité qu'il demande, et un deuxième à appliquer aux termes dont la testabilité a déjà été testée, noté $\{Y/\theta X\}_t$, et qui omet ce qui est censé avoir été déjà enregistré (à la fois les étiquettes visitées et les tests de testabilité redondants). C'est dans les deux cas la sixième équation (celle gérant la décomposition d'une application) qui montrera l'articulation entre ces deux modes.

Définition 6.41 (Filtrage composé étiqueté). *L'expression $\alpha : (\Delta, \mu)$ (resp. $\Gamma \cdot (\Delta, \mu)$) est une notation pour la paire $(\alpha\Delta, \mu)$ (resp. $(\Gamma\Delta, \mu)$). Dans l'expression $\{Y/\theta X\}_m$ utilisée ci-dessous, le mode m désigne soit d soit t . Ces deux modes sont mutuellement définis ainsi :*

$$\begin{aligned} \{Y/\theta \alpha : Z\}_d &= \alpha : \{Y/\theta Z\}_d \\ \{Y/\theta \hat{x}\}_d &= (\varepsilon, \{x := Y\}) && x \in \theta \\ \{Y/\theta ((\tau)P.B)\}_d &= (\varepsilon, \perp) \\ \{\alpha : Z/\theta X\}_d &= \alpha : \{Z/\theta X\}_d && \text{sinon} \\ \{\hat{x}/_{\theta} \hat{x}\}_d &= (\varepsilon, \{\}) && x \notin \theta \\ \{A_1 A_2 /_{\theta} P_1 P_2\}_d &= |P_1| \cdot |A_1| \cdot (\{A_1/\theta P_1\}_t \uplus \{A_2/\theta P_2\}_d) && A_1 A_2, P_1 P_2 \text{ formes testables} \\ \{Y/\theta X\}_d &= (|X| \cdot |Y|, \perp) && Y, X \text{ formes testables, sinon} \\ \{Y/\theta X\}_d &= \bar{X} && \text{sinon} \\ \\ \{Y/\theta \alpha : Z\}_t &= \{Y/\theta Z\}_t \\ \{Y/\theta \hat{x}\}_t &= (\varepsilon, \{x := Y\}) && x \in \theta \\ \{Y/\theta ((\tau)P.B)\}_t &= (\varepsilon, \perp) \\ \{\alpha : Z/\theta X\}_t &= \{Z/\theta X\}_t && \text{sinon} \\ \{\hat{x}/_{\theta} \hat{x}\}_t &= (\varepsilon, \{\}) && x \notin \theta \\ \{A_1 A_2 /_{\theta} P_1 P_2\}_t &= \{A_1/\theta P_1\}_t \uplus \{A_2/\theta P_2\}_d && A_1 A_2, P_1 P_2 \text{ formes testables} \\ \{Y/\theta X\}_t &= (\varepsilon, \perp) && Y, X \text{ formes testables, sinon} \\ \{Y/\theta X\}_t &= \bar{X} && \text{sinon} \end{aligned}$$

où l'union disjointe de filtrats \uplus est étendue aux paires par la table suivante :

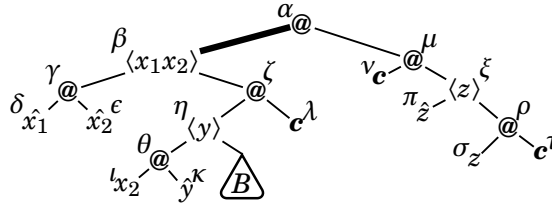
\uplus	(Δ_2, σ_2)	(Δ_2, \perp)	\bar{X}
(Δ_1, σ_1)	$(\Delta_1 \Delta_2, \sigma_1 \uplus \sigma_2)$	(Δ_2, \perp)	\bar{X}
(Δ_1, \perp)	(Δ_1, \perp)	$(\Delta_1 \Delta_2, \perp)$	(Δ_1, \perp)
\bar{X}	\bar{X}	(Δ_2, \perp)	\bar{X}

Fait à noter dans la forme de cette union disjointe : la séquence Δ associée à une branche dans laquelle une erreur de filtrage a été constatée est toujours conservée. En particulier, une erreur de filtrage l'emporte (et transmet son Δ) sur une branche indéfinie, et la combinaison de deux erreurs retient les deux séquences Δ . En revanche, une erreur combinée à un filtrage réussi efface à la fois le résultat du filtrage réussi et le Δ associé (pour ne garder que le Δ de la branche produisant l'erreur).

Remarque 6.42. *Le mode $\{\{Y/\theta X\}_t\}$ étant censé être utilisé uniquement sur des motifs et arguments dont la testabilité a déjà été testée, les conditions de testabilité accompagnant l'avant-dernière et l'antépénultième équations sont inutiles. De même, en utilisation normale le cas Σ est superflu pour ce mode.*

Exemple 6.43.

Pour le radical du terme étiqueté (L1) (exemple 6.32) représenté graphiquement par



le filtrage composé étiqueté simple réussit et renvoie la substitution

$$\sigma = \{x_1 := \nu : \mathbf{c}, x_2 := \xi : (\langle z \rangle \pi : \hat{z} . \rho : (\sigma : z)(\tau : \mathbf{c}))\}$$

ainsi que la séquence d'étiquettes

$$\Delta = \gamma \delta \mu \nu \epsilon$$

En préparation de la section 6.6 sur les propriétés causales du système étiqueté que nous définissons ici, le lemme 6.44 énonce une propriété du filtrage composé étiqueté par défaut $\{\{Y/\theta X\}_d\}$ qui le différencie de l'autre version $\{\{Y/\theta X\}_t\}$.

Lemme 6.44 (Témoins de testabilité). *Soient X et Y deux termes étiquetés. Si $\{\{Y/\theta X\}_d = (\Delta, \mu)$, alors $|X| \sqsubseteq \Delta$ et $|Y| \sqsubseteq \Delta$.*

Démonstration. Par induction sur $\{\{Y/\theta X\}_d$. □

Comme dans le *Pure Pattern Calculus*, le résultat du *filtrage* d'un argument étiqueté par un motif étiqueté est obtenu après un test sur le domaine de la substitution éventuelle renvoyée par le filtrage composé étiqueté, en ajoutant que le filtrage composé étiqueté commence en mode « défaut ».

Définition 6.45 (Filtrage étiqueté). *Le **filtrage étiqueté** d'un argument étiqueté A par un motif étiqueté P avec variables de filtrage θ , noté $\{A/\theta P\}$, est défini par les cas suivants :*

- Si $\{A/\theta P\}_d = (\Delta, \perp)$, alors $\{A/\theta P\} = (\Delta, \perp)$.
- Si $\{A/\theta P\}_d = (\Delta, \sigma)$ et $\theta = \text{dom}(\sigma)$, alors $\{A/\theta P\} = (\Delta, \sigma)$.
- Si $\{A/\theta P\}_d = (\Delta, \sigma)$ et $\theta \neq \text{dom}(\sigma)$, alors $\{A/\theta P\} = (\Delta, \perp)$.

Remarquons que le filtrage étiqueté de A par P n'est pas défini si $\{A/\theta P\}_d = \bar{X}$.

6.5 Pure Pattern Calculus faible étiqueté

Nous pouvons maintenant définir la dynamique du *Pure Pattern Calculus* faible étiqueté et montrer que le système définit un système de partage étiqueté *SLS* essentiellement de la même façon qu'à la section 5.3 pour le λ -calcul faible étiqueté.

6.5.1 Définition

Comme dans les chapitres 2, 3 et 5 la réduction étiquetée va isoler un préfixe de la fonction dont les étiquettes seront modifiées. Au chapitre 2 le préfixe modifié était le squelette de la fonction (pour modéliser la technique de Wadsworth [Wad71]), tandis qu'au chapitre 5 il s'agissait de son épine (pour rester au plus près du λ -calcul étiqueté de Blanc, Lévy et Maranget [BLM07]). Nous avons vu au chapitre 3 que les deux options généraient pour le λ -calcul la même réduction partagée (théorème 3.26 de partage équivalent). De plus nous avons évoqué à la section 5.6 que les deux options pourraient avoir la même pertinence du point de vue de la causalité, la propagation des étiquettes au squelette ne faisant pas plus qu'anticiper des propagations qui sont faites à terme dans le λ -calcul de Blanc, Lévy et Maranget [BLM07] (nous verrons au chapitre 7 dans le cadre des *CRS* faibles orthogonaux qu'utiliser les squelettes comme base des propagations d'étiquettes permet effectivement d'obtenir les mêmes résultats de correction et de complétude causales).

Nous avons donc la liberté pour cette section de choisir entre les épines et les squelettes pour définir les zones dont les étiquettes sont modifiées. Nous prendrons les squelettes, dont la définition va maintenant être adaptée à *PPC*. Comme pour le λ -calcul (définition 2.5) un squelette $\langle\langle t \rangle\rangle^\theta$ est obtenu en retirant de t ses expressions libres dans lesquelles aucune variable de θ n'apparaît libre. Notons que dans la transcription à *PPC* nous parlons bien des variables libres, et non des *matchables* libres.

Définition 6.46 (Theta-squelettes). *Le **θ -squelette** d'un terme t , noté $\langle\langle t \rangle\rangle^\theta$, est le préfixe de t défini ainsi :*

$$\begin{aligned} \langle\langle t \rangle\rangle^\theta &= \square & \theta \cap \text{fv}(t) &= \emptyset \\ \text{Sinon :} & & & \\ \langle\langle x \rangle\rangle^\theta &= x & x &\in \theta \\ \langle\langle t_1 t_2 \rangle\rangle^\theta &= \langle\langle t_1 \rangle\rangle^\theta \langle\langle t_2 \rangle\rangle^\theta \\ \langle\langle \tau p.b \rangle\rangle^\theta &= \langle\tau \rangle \langle\langle p \rangle\rangle^\theta . \langle\langle b \rangle\rangle^{\theta \cup \tau} \end{aligned}$$

Le **squelette** d'une abstraction $\langle \theta \rangle p.b$ est le préfixe $\langle \theta \rangle \square. \langle \langle b \rangle \rangle^\theta$.

Comme ici les étiquettes augmentent la structure des termes, précisons la définition des squelettes sur les termes du *Pure Pattern Calculus* étiqueté :

Définition 6.47 (Theta-squelettes étiquetés). *Le θ -squelette étiqueté d'un terme étiqueté X , noté $\langle \langle X \rangle \rangle^\theta$, est le préfixe de X défini ainsi :*

$$\begin{aligned} \langle \langle X \rangle \rangle^\theta &= \square & \theta \cap \text{fv}(X) &= \emptyset \\ \text{Sinon :} \\ \langle \langle \alpha : X \rangle \rangle^\theta &= \alpha : \langle \langle X \rangle \rangle^\theta \\ \langle \langle x \rangle \rangle^\theta &= x & x &\in \theta \\ \langle \langle T_1 T_2 \rangle \rangle^\theta &= (\langle \langle T_1 \rangle \rangle^\theta) (\langle \langle T_2 \rangle \rangle^\theta) \\ \langle \langle \tau \rangle P.B \rangle^\theta &= \langle \tau \rangle \langle \langle P \rangle \rangle^\theta. \langle \langle B \rangle \rangle^{\theta \cup \tau} \end{aligned}$$

Le **squelette** d'une abstraction $\langle \theta \rangle P.B$ est le préfixe $\langle \theta \rangle \square. \langle \langle B \rangle \rangle^\theta$.

De même, le réétiquetage uniforme $[\Omega, X]$ d'un contexte d'arité arbitraire X se transpose encore au cadre de *PPC* :

$$\begin{aligned} [\Omega, \alpha : X] &= [\Omega, \alpha] : [\Omega, X] \\ [\Omega, \square] &= \square \\ [\Omega, x] &= x \\ [\Omega, \hat{x}] &= \hat{x} \\ [\Omega, T_1 T_2] &= [\Omega, T_1] [\Omega, T_2] \\ [\Omega, \langle \theta \rangle P.B] &= \langle \theta \rangle [\Omega, P]. [\Omega, B] \end{aligned}$$

Nous pouvons enfin définir la réduction du *Pure Pattern Calculus* faible étiqueté.

Définition 6.48 (Règles de réduction étiquetée). *Un **pré-radical étiqueté** est un terme étiqueté complet de la forme*

$$R = \alpha : ((\Gamma \cdot \langle \theta \rangle P.B)A)$$

*Un tel pré-radical étiqueté est un **radical** quand à la fois sa position n'est pas gelée et le filtrage $\{A/\theta P\}$ est défini. Dans ce cas, notons $\{A/\theta P\} = (\Delta, \mu)$ et attribuons le nom $\Omega = \alpha \Gamma \Delta$ à tout pas de réduction dont le radical est R . Notons de plus $B = S[B_1, \dots, B_n]$ avec $S = \langle \langle B \rangle \rangle^\theta$. Nous avons alors*

$$R = \alpha : ((\Gamma \cdot \langle \theta \rangle P.S[B_1, \dots, B_n])A) \xrightarrow{R} [\Omega] : ([\Omega, S][B_1, \dots, B_n])^\mu$$

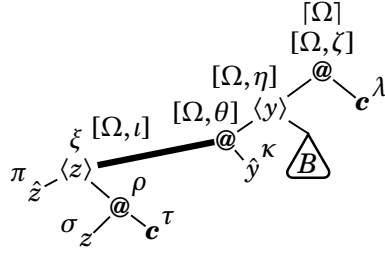
où comme précédemment $X^\perp = T_\perp$ pour tout terme X , avec un terme T_\perp fixé arbitraire qui est une forme normale close, par exemple l'identité munie de certaines étiquettes initiales distinguées :

$$T_\perp = \alpha_\perp : \langle x \rangle (\pi_\perp : \hat{x}). (\beta_\perp : x)$$

C'est ce dernier cas d'échec $X^\perp = t_\perp$ qui motive l'utilisation de l'expression $([\Omega, S][B_1, \dots, B_n])^\mu$ en place de l'expression $[\Omega, S]^\mu[B_1, \dots, B_n]$ plus couramment utilisée dans les chapitres précédents, mais qui ne serait pas correcte ici.

Exemple 6.49.

La réduction du radical (L1) de l'exemple 6.32 donne le réduit suivant, avec $\Omega = \alpha\beta\gamma\delta\mu\nu\epsilon$.



6.5.2 Propriété de partage

Nous vérifions maintenant que le *Pure Pattern Calculus* faible étiqueté est un *SLS*, et donc spécifie une notion de réduction partagée. La vérification reprend essentiellement celle faite au chapitre 2 pour le λ -calcul faible étiqueté spécifiant la technique de Wadsworth.

Comme au chapitre 5, l'ordre sur les étiquettes utilisé pour la définition d'un *SLS* est adapté à la présence de séquence d'étiquettes, et ne coïncide pas avec la relation de contribution \hookrightarrow .

Définition 6.50 (Ordre sur les étiquettes). *L'ensemble \mathcal{L} des étiquettes du Pure Pattern Calculus faible étiqueté est muni de l'ordre partiel \leq généré par :*

$$\begin{aligned} \omega_1 &\leq [\omega_1 \dots \omega_n] \\ \omega_1 &\leq [\omega_1 \dots \omega_n, \alpha] \end{aligned}$$

Comme au chapitre 2 (définition 2.22 et lemme 2.24) nous allons utiliser l'invariant des *squelettes arborescents*, qui témoigne du fait qu'aucun partage ne peut apparaître à l'intérieur d'un squelette en réduction faible.

Lemme 6.51 (Squelettes et substitutions, adapté du lemme 2.7). *Soient $\langle \theta \rangle p.b$ un cas et σ une substitution telles que $(\langle \theta \rangle p.b)^\sigma$ est définie. Alors $\langle \theta \rangle p.b$ et $(\langle \theta \rangle p.b)^\sigma$ ont même squelette.*

Démonstration. Si $\text{dom}(\sigma) \cap \text{fv}(\langle \theta \rangle p.b) = \emptyset$ alors c'est immédiat. Supposons donc $\text{dom}(\sigma) \cap \text{fv}(\langle \theta \rangle p.b) \neq \emptyset$. Les occurrences libres de variables du domaine de σ sont en particulier des expressions libres de $\langle \theta \rangle p.b$ et n'appartiennent pas au squelette. De même, la définition de $(\langle \theta \rangle p.b)^\sigma$ implique que les occurrences substituées d'éléments du codomaine de σ sont libres dans $(\langle \theta \rangle p.b)^\sigma$ et n'influent pas sur le squelette. \square

Lemme 6.52 (Invariance des squelettes, adapté du lemme 2.8). *Soient $\rho : t \rightarrow t'$ une étape de réduction faible et s' un squelette dans t' de position q' . Notons q l'ancêtre dans t de q' . Alors $t|_q$ est une abstraction et son squelette s est égal à s' .*

Démonstration. Par cas sur la position $q_\rho = \text{root}(\rho)$ de ρ relativement à q .

- Si q_ρ et q sont disjointes alors c'est immédiat.
- Si q est un préfixe de q_ρ , alors q_ρ est soit dans le squelette, soit dans une expression libre de $t|_q$. Le premier cas est interdit par la réduction faible car il implique que q_ρ soit gelée. Dans le deuxième cas le lemme 6.51 conclut.
- Si q_ρ est un préfixe de q , alors soit s est le squelette principal du radical, soit s est strictement inclus dans ce dernier, soit s est inclus dans une expression libre de $t|_{q_\rho}$. Le premier cas contredit l'hypothèse car la racine du squelette principal n'a pas de descendant. Le deuxième cas est résolu par le lemme 6.51. Dans le troisième cas $t|_q$ est égal à ses descendants éventuels. \square

Nous pouvons obtenir immédiatement l'équivalent étiqueté :

Lemme 6.53 (Invariance des squelettes, adapté du lemme 2.23). *Soit $X \rightarrow X'$ une étape de réduction étiquetée faible dont le radical a pour nom Ω , et soit S' un squelette dans X' . Notons S le squelette de X ancêtre de S' . Alors $S' = S$ ou $S' = [\Omega, S]$.*

Définition 6.54 (Propriété des squelettes arborescents). *Nous dirons qu'un terme étiqueté X vérifie la propriété des squelettes arborescents, noté $\mathbb{F}(X)$ si pour tout squelette S dans X , les étiquettes de S sont deux à deux distinctes.*

Le lemme 6.53 a pour corollaire immédiat :

Lemme 6.55 (Squelette arborescents). *Soit $X \rightarrow X'$ une étape de réduction étiquetée faible. Si $\mathbb{F}(X)$ alors $\mathbb{F}(X')$.*

Enfin, nous pouvons procéder à l'instanciation du formalisme abstrait des SLS sur le *Pure Pattern Calculus* faible étiqueté. Dans cette instanciation nous notons $\text{seq}(\mathcal{X})$ l'ensemble des séquences sur \mathcal{X} .

Instanciation 6.56. *Considérons le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- la signature $\mathcal{S} = \mathcal{X} \cup \{\hat{x} \mid x \in \mathcal{X}\} \cup \mathcal{L} \cup \{\text{@}\} \cup \{[\theta] \mid \theta \in \text{seq}(\mathcal{X})\}$, où les étiquettes sont considérées comme des symboles d'arité 1 et les cas comme des symboles d'arité 2.
- l'ensemble \mathcal{T} des termes étiquetés vérifiant la propriété des squelettes arborescents.
- l'ensemble \mathcal{R} des pas de réduction faible étiquetée dont la source vérifie la propriété des squelettes arborescents.

Pour tout pas de réduction $\rho \in \mathcal{R}$, notons

$$c[\alpha : ((\Gamma \cdot \langle \theta \rangle P.S[B_1, \dots, B_n])A)] \rightarrow c[[\Omega] : ((\Omega, S)[B_1, \dots, B_n])^\mu]$$

et définissons de même que précédemment :

- $\text{dex}(\rho) = \alpha : ((\Gamma \cdot \langle \theta \rangle P.S[B_1, \dots, B_n])A)$
 - $\text{duct}(\rho) = [\Omega] : ([\Omega, S][B_1, \dots, B_n])^\mu$
 - $\text{ctx}(\rho) = c$
 - $\text{effz}(\rho) = \{c\} \cup 1 \cdot (\text{pos}(S) \setminus \text{fo}_\theta(S))$
- où $\text{fo}_\theta(S) = \bigcup_{x \in \theta} \text{fo}_x(S)$.

Lemme 6.57. *Le système $\Sigma = (\mathcal{L}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration.

Source & Cible Par lemme 6.55.

Résidus L'ancêtre d'une position gelée est gelé. De plus la méta-opération de filtrage n'est définie qu'en fonction du radical et non de son contexte.

Zone d'effet Soit un pas de réduction

$$c[\alpha : ((\Gamma \cdot \langle \theta \rangle P.S[B_1, \dots, B_n])A)] \rightarrow c[[\Omega] : ([\Omega, S][B_1, \dots, B_n])^\mu]$$

Par définition de $\text{effz}(\rho)$, toute position hors de la zone d'effet désigne soit un sous-terme de A soit un sous-terme de l'un des B_i . \square

Lemme 6.58. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine La racine de tout radical est une étiquette.

Progression Toute étiquette de la zone d'effet d'un pas de réduction est de la forme $[\Omega]$ ou $[\Omega, \alpha]$ où Ω est le nom du radical. Par définition, Ω est une séquence d'étiquettes commençant par $\tau(\rho)$. Donc en particulier $\tau(\rho) \leq [\Omega]$ et $\tau(\rho) \leq [\Omega, \alpha]$.

Héritage Soient un pas de réduction $\rho \in \mathcal{R}$ et deux étiquettes $\alpha, \beta \in \mathcal{L}$ telles que β est une étiquette de la zone d'effet de ρ , $\alpha \neq \tau(\rho)$ et $\alpha < \beta$. En particulier β est de la forme $[\omega_1 \dots \omega_n]$ ou $[\omega_1 \dots \omega_n, \gamma]$ avec $\omega_1 = \tau(\rho)$. Donc $\alpha \leq \omega_1$ et même $\alpha < \omega_1$.

Partage Si on omet sa racine qui est une étiquette $[\Omega]$, la zone d'effet d'un pas de réduction est de la forme $[\Omega, S]$ où S est inclus dans un squelette de la source. Par propriété des squelette arborescents, les étiquettes de deux positions différentes de la zone d'effet ne peuvent donc être égales. \square

Ainsi, notre étiquetage du *Pure Pattern Calculus* faible définit un SLS et donc une notion de réduction partagée. Nous allons maintenant discuter les propriétés causales de cet étiquetage.

6.6 Propriétés causales

L'enjeu de cette section est la preuve des lemmes 6.59 de contribution directe et 6.60 de stabilité des radicaux, qui expriment certaines propriétés causales de l'étiquetage du *Pure Pattern Calculus* proposé en section 6.5.

Le lemme 6.59 de contribution directe est l'exact équivalent du lemme 5.22 de contribution directe énoncé pour le λ -calcul faible étiqueté. Le lemme 6.60 de stabilité des radicaux en revanche diffère de son homologue 5.23 : comme nous le verrons à la section 6.7 il n'est plus possible dans le *Pure Pattern Calculus* d'assurer que le nom d'un résidu est toujours égal au nom de son ancêtre comme nous l'avions fait dans le lemme 5.23 pour le λ -calcul faible étiqueté. À la place, le lemme 6.60 énonce une certaine monotonie dans l'évolution des noms de radicaux par résiduation : l'ensemble des contributeurs du nom d'un radical ne peut que croître par résiduation. Les modalités de cette croissance sont énoncées précisément par les lemmes 6.70 et 6.71 de stabilité des filtrages.

Les propriétés de correction et de complétude causales qui étaient énoncées par les théorèmes 5.27 et 5.29 pour le λ -calcul faible étiqueté ne seront pas reproduites ici, mais seront discutées à la section 6.7.

Énonçons d'abord les deux lemmes cibles de cette section.

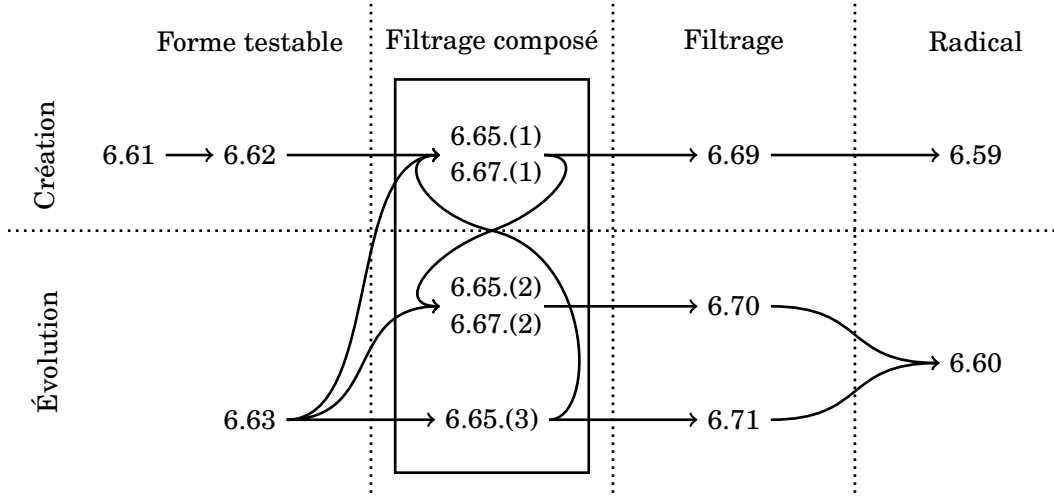
Lemme 6.59 (Contribution directe). *Pour tout pas de réduction $\rho : T \rightarrow T'$ de nom Ω , si ρ_c est un pas de réduction de source T' et de nom Ω_c qui est créé par ρ , alors $\Omega \hookrightarrow^d \Omega_c$.*

Lemme 6.60 (Stabilité des radicaux). *Soit $\rho : X \rightarrow X'$ un pas de réduction de nom Ω . Soit ρ_a un pas de réduction de source X et de nom Ω_a et de radical R_a . Si R_d est un descendant de R_a dans X' , alors R_d est le radical d'un pas de réduction ρ_d (résidu de ρ_a) et de nom Ω_d tel que :*

- Si $\Omega_a \neq \Omega_d$ alors $\Omega \hookrightarrow^d \Omega_d$.
- Pour tout nom Ω' tel que $\Omega' \hookrightarrow \Omega_a$ nous avons aussi $\Omega' \hookrightarrow \Omega_d$.

Pour chacun de ces deux lemmes 6.59 et 6.60, l'étape intermédiaire principale sera une étude du résultat d'une opération de filtrage étiqueté, avec respectivement le lemme 6.69 de réduction dans un filtrage indéfini et les lemmes 6.70 et 6.71 de réduction dans un filtrage défini. Ces trois lemmes 6.69, 6.70 et 6.71 seront essentiellement issus d'une analyse de l'opération de filtrage composé étiqueté, dans laquelle ces deux aspects (réduction dans un filtrage indéfini ou défini) devront être mélangés, à cause de l'entrailecement entre filtrages définis et indéfinis qui est apporté par la définition de l'union disjointe de filtrats (définition 6.41). C'est le lemme 6.65 du filtrage composé étiqueté et son jumeau 6.67 qui vont sanctionner cette analyse de toutes les évolutions possibles d'un filtrage composé étiqueté (défini ou indéfini) et représenter la majeure partie du travail technique de cette section.

L'ensemble est résumé dans le graphe de dépendances suivant.



Un premier groupe de résultats intermédiaires est formé par les lemmes 6.61 à 6.63 qui donnent les propriétés de création et de stabilité des formes testables. Les deux résultats principaux sont la stabilité des témoins testabilité par descendance (lemme 6.63) et l'apparition du nom du pas de réduction dans le témoin de testabilité de toute nouvelle forme testable (lemme 6.62).

Lemme 6.61 (Création d'une structure de données). *Soient X un terme étiqueté qui n'est pas une structure de données étiquetée, et $\rho : X \rightarrow X'$ un pas de réduction de nom Ω tel que X' est une structure de données étiquetée. Alors $[\Omega] \in |X'|$.*

Démonstration. Raisonnons par induction sur $\rho : X \rightarrow X'$ avec X' une structure de données étiquetée mais pas X :

- Cas $\alpha : ((\Gamma \cdot \langle \theta \rangle P.B)A) \rightarrow X'$ par un pas de réduction à la racine. Alors X' est de la forme $[\Omega] : B'$. En particulier $|X'| = [\Omega] |X|$ et $[\Omega] \in |X'|$.
- Cas $\alpha : Z \rightarrow \alpha : Z'$ avec $Z \rightarrow Z'$. Le terme étiqueté $\alpha : Z$ n'est pas une structure de données étiquetée, ainsi Z n'est pas une structure de données étiquetée. De même $\alpha : Z'$ est une structure de données étiquetée, ainsi Z' est une structure de données étiquetée. Alors par hypothèse d'induction $[\Omega] \in |Z'|$. Donc $[\Omega] \in |\alpha : Z'|$.
- Cas $T_1 T_2 \rightarrow T'_1 T_2$ avec $T_1 \rightarrow T'_1$. Comme précédemment, T'_1 est une structure de données étiquetées mais pas T_1 . Par hypothèse d'induction $[\Omega] \in |T'_1|$. Donc $[\Omega] \in |T'_1 T_2|$.
- Cas $T_1 T_2 \rightarrow T_1 T'_2$ avec $T_2 \rightarrow T'_2$. Comme précédemment T_1 n'est pas une structure de données étiquetée. En particulier $T_1 T'_2$ n'est pas non plus une structure de données étiquetée, ce qui n'est pas notre cas.
- Cas $\langle \theta \rangle P.B \rightarrow X'$. Alors X' est de la forme $\langle \theta \rangle P'.B'$ et en particulier n'est pas une structure de données étiquetée, ce qui n'est pas notre cas. \square

Lemme 6.62 (Création d'une forme testable). *Soient X un terme étiqueté qui n'est pas une forme testable, et $\rho : X \rightarrow X'$ un pas de réduction de nom Ω tel que X' est une forme testable. Alors $[\Omega] \in |X'|$.*

Démonstration. Si X' est une structure de données, alors le lemme 6.61 conclut. Supposons donc que X' est de la forme $\Gamma' \cdot ((\theta')P'.B')$. Dans ce cas $|X'| = \Gamma'$. Par hypothèse X n'est pas une forme testable, et est donc de la forme $\Gamma \cdot R$ avec R le radical de ρ . Le réduit de ρ est de la forme $[\Omega] : R'$. En particulier $X' = \Gamma \cdot ([\Omega] : R')$, donc $\Gamma[\Omega]$ est un préfixe de $|X'|$ et $[\Omega] \in |X'|$. \square

Lemme 6.63 (Stabilité de la testabilité). *Soient X un terme étiqueté qui est une forme testable, et $\rho : X \rightarrow X'$ un pas de réduction. Alors X' est toujours une forme testable, et $|X'| = |X|$.*

Démonstration. Par induction sur la définition des formes testables, en remarquant qu'un pré-radical n'est pas une forme testable. \square

Nous arrivons maintenant au niveau du filtrage composé étiqueté, où les résultats de création d'un filtrage composé et d'évolution d'un filtrage composé qui réussit ou qui échoue doivent être associés dans une même analyse (voir remarque 6.66 entre l'énoncé et la démonstration du lemme 6.65). Nous séparons en un lemme 6.65 relatif à la réduction dans l'argument et un lemme 6.67 relatif à la réduction dans le motif.

Convention 6.64. *Soient deux substitutions $\sigma_T = \{x_1 := T_1, \dots, x_n := T_n\}$ et $\sigma_U = \{x_1 := U_1, \dots, x_n := U_n\}$ qui ont même support. Nous noterons $\sigma_T \rightarrow \sigma_U$ s'il existe $i \in \{1, \dots, n\}$ tel que $T_i \rightarrow U_i$ et pour tout $j \neq i$, $T_j = U_j$.*

Rappelons qu'un symbole μ représente un filtrat défini, c'est-à-dire une substitution σ ou un échec \perp .

Lemme 6.65 (Filtrage composé étiqueté). *Soient X et Y deux termes étiquetés, $\rho : Y \rightarrow Y'$ un pas de réduction de nom Ω , et $m \in \{d, t\}$. Les trois propositions suivantes sont vérifiées :*

- (1) *Si $\{\{Y/\theta X\}\}_m = \bar{X}$ et $\{\{Y'/\theta X\}\}_m = (\Delta', \mu')$, alors :*
 - (a) *Soit $m = d$ et $[\Omega] \in \Delta'$.*
 - (b) *Soit $m = t$ et $[\Omega] \in |Y'| \cdot \Delta'$.*
- (2) *Si $\{\{Y/\theta X\}\}_m = (\Delta, \perp)$, alors $\{\{Y'/\theta X\}\}_m = (\Delta', \perp)$, avec trois cas :*
 - (a) *Soit $\Delta = \Delta'$.*
 - (b) *Soit $\Delta \sqsubseteq \Delta'$, $m = d$, et $[\Omega] \in \Delta'$.*
 - (c) *Soit $\Delta \sqsubseteq \Delta'$, $m = t$, et $[\Omega] \in |Y'| \cdot \Delta'$.*
- (3) *Si $\{\{Y/\theta X\}\}_m = (\Delta, \sigma)$, alors $\{\{Y'/\theta X\}\}_m = (\Delta, \sigma')$ avec $\sigma \rightarrow \sigma'$.*

Remarque 6.66. *Les trois propositions du lemme 6.65 vont être prouvées dans une même induction. En effet, le cas d'analyse correspondant à la décomposition d'une application et à la combinaison des résultats (par l'opérateur \oplus) va nous demander de combiner les aspects de création d'un nouveau filtrage et d'évolution d'un filtrage déjà défini (qu'il soit une réussite ou un échec). En particulier, le cas VI)ii) « \bar{X} contre \perp » utilisera en hypothèse la proposition (1) pour vérifier la proposition (2). Également, le cas VI)vii) « σ contre \bar{X} » utilisera en hypothèse la*

proposition (3) pour invalider les prémisses de la proposition (1). Les cas VI)viii) et VI)ix) enfin montrerons une dépendance directe de (3) vers (2).

Ainsi, la séparation d'un côté des résultats de création (1) et de l'autre côté des résultats d'évolution (2) et (3) n'est pas possible puisque (2) dépend de (1), qui lui-même dépend de (3). Ces dépendances reflètent le fait que l'opération \uplus donne la priorité à \perp sur \bar{X} et σ et à \bar{X} sur σ . Il aurait donc en revanche été possible de prouver d'abord (3), puis (1), et enfin (2).

Démonstration du lemme 6.65. Nous prouvons les trois propositions dans une même induction sur la définition de $\{\{Y/\theta X\}_m\}$. Nous avons à considérer les huit équations de la définition du filtrage composé étiqueté.

- I) Cas $\{\{Y/\theta \alpha : Z\}_m = \alpha^? : \{\{Y/\theta Z\}_m\}$, où $\alpha^?$ dénote l'étiquette α ou une absence d'étiquette en fonction de $m \in \{d, t\}$. Raisonnons par cas sur $\{\{Y/\theta Z\}_m\}$.
- Cas $\{\{Y/\theta Z\}_m = \bar{X}$. Alors $\{\{Y/\theta \alpha : Z\}_m = \bar{X}$. Raisonnons par cas sur $\{\{Y'/\theta Z\}_m\}$.
 - Cas $\{\{Y'/\theta Z\}_m = \bar{X}$. Alors $\{\{Y'/\theta \alpha : Z\}_m = \bar{X}$, et il n'y a rien à vérifier.
 - Cas $\{\{Y'/\theta Z\}_m = (\Delta', \mu)$. Alors $\{\{Y'/\theta \alpha : Z\}_m = (\alpha^? \Delta', \mu)$, et nous vérifions la proposition (1) sur $\{\{Y'/\theta \alpha : Z\}_m\}$. L'hypothèse (1) sur $\{\{Y'/\theta Z\}_m\}$ prévoit deux cas :
 - (a) Cas $m = d$, $\alpha^? = \alpha$, et $[\Omega] \in \Delta'$. Alors $[\Omega] \in \alpha \Delta'$ et la proposition (1.a) est vérifiée.
 - (b) Cas $m = t$, $\alpha^? = \cdot$, et $[\Omega] \in |Y'| \cdot \Delta'$. Alors $[\Omega] \in |Y'| \cdot \Delta'$ et la proposition (1.b) est vérifiée.
 - Cas $\{\{Y/\theta Z\}_m = (\Delta, \perp)$. Nous vérifions la proposition (2) sur $\{\{Y'/\theta \alpha : Z\}_m\}$. Par hypothèse (2) sur $\{\{Y'/\theta Z\}_m\}$ nous avons $\{\{Y'/\theta Z\}_m = (\Delta', \perp)$, donc $\{\{Y'/\theta \alpha : Z\}_m = (\alpha^? \Delta', \perp)$. L'hypothèse (2) prévoit de plus trois cas :
 - (a) Cas $\Delta' = \Delta$. En particulier $\alpha^? \Delta' = \alpha^? \Delta$, donc la proposition (2.a) est vérifiée.
 - (b) Cas $\Delta \sqsubseteq \Delta'$, $m = d$, $\alpha^? = \alpha$ et $[\Omega] \in \Delta'$. En particulier $\alpha \Delta \sqsubseteq \alpha \Delta'$ et $[\Omega] \in \alpha \Delta'$, donc la proposition (2.b) est vérifiée.
 - (c) Cas $\Delta \sqsubseteq \Delta'$, $m = t$, $\alpha^? = \cdot$, et $[\Omega] \in |Y'| \cdot \Delta'$. En particulier $\Delta \sqsubseteq \Delta'$ et $[\Omega] \in |Y'| \cdot \Delta'$, donc la proposition (2.c) est vérifiée.
 - Cas $\{\{Y/\theta Z\}_m = (\Delta, \sigma)$. Alors $\{\{Y/\theta \alpha : \Delta\}_m = (\alpha^? \Delta, \sigma)$, et nous vérifions la proposition (3) sur $\{\{Y'/\theta \alpha : Z\}_m\}$. Par hypothèse (3) sur $\{\{Y'/\theta Z\}_m\}$ nous avons $\{\{Y'/\theta Z\}_m = (\Delta, \sigma')$, donc $\{\{Y'/\theta \alpha : Z\}_m = (\alpha^? \Delta, \sigma')$ avec en particulier $\alpha^? \Delta = \alpha^? \Delta$. L'hypothèse (3) donne de plus $\sigma \rightarrow \sigma'$, donc la proposition (3) est vérifiée.
- II) Cas $\{\{Y/\theta \hat{x}\}_m = (\varepsilon, \{x := Y\})$ avec $x \in \theta$. Nous vérifions la proposition (3) sur $\{\{Y'/\theta \hat{x}\}_m\}$. Par définition $\{\{Y'/\theta \hat{x}\}_m = (\varepsilon, \{x := Y'\})$ avec $\varepsilon = \varepsilon$ et $\{x := Y\} \rightarrow \{x := Y'\}$, donc la proposition (3) est vérifiée.
- III) Cas $\{\{Y/\theta \langle \tau \rangle P.B\}_m = (\varepsilon, \perp)$. Nous vérifions la proposition (2) sur $\{\{Y'/\theta \langle \tau \rangle P.B\}_m\}$. Par définition $\{\{Y'/\theta \langle \tau \rangle P.B\}_m = (\varepsilon, \perp)$ avec $\varepsilon = \varepsilon$, donc la proposition (2.a) est vérifiée.

- IV) Cas $\{\alpha : Z/\theta X\}_m = \alpha^? : \{Z/\theta X\}_m$, sinon, où comme ci-dessus $\alpha^?$ dénote l'étiquette α ou une absence d'étiquette en fonction de $m \in \{v, t\}$. Il y a deux cas à considérer en fonction de la position de la réduction.
- i) Cas $\alpha : Z \rightarrow \alpha : Z'$ avec $Z \rightarrow Z'$. Raisonnons par cas sur $\{Z/\theta X\}_m$.
- Cas $\{Z/\theta X\}_m = \bar{X}$. Alors $\{\alpha : Z/\theta X\}_m = \bar{X}$. Raisonnons par cas sur $\{Z'/\theta X\}_m$.
 - Cas $\{Z'/\theta X\}_m = \bar{X}$. Alors $\{\alpha : Z'/\theta X\}_m = \bar{X}$, et il n'y a rien à vérifier.
 - Cas $\{Z'/\theta X\}_m = (\Delta', \mu)$. Alors $\{\alpha : Z'/\theta X\}_m = (\alpha^? \Delta', \mu)$, et nous vérifions la proposition (1) sur $\{\alpha : Z'/\theta X\}_m$. L'hypothèse (1) sur $\{Z'/\theta X\}_m$ prévoit deux cas :
 - (a) Cas $m = d$, $\alpha^? = \alpha$, et $[\Omega] \in \Delta'$. Alors $[\Omega] \in \alpha \Delta'$ et la proposition (1.a) est vérifiée.
 - (b) Cas $m = t$, $\alpha^? = \cdot$, et $[\Omega] \in |Z'| \cdot \Delta'$. Alors $[\Omega] \in |\alpha : Z'| \cdot \Delta'$ et la proposition (1.b) est vérifiée.
 - Cas $\{Z/\theta X\}_m = (\Delta, \perp)$. Alors $\{\alpha : Z/\theta X\}_m = (\alpha^? \Delta, \perp)$, et nous vérifions la proposition (2) $\{\alpha : Z/\theta X\}_m$. Par hypothèse (2) sur $\{Z/\theta X\}_m$ nous avons $\{Z'/\theta X\}_m = (\Delta', \perp)$, donc $\{\alpha : Z'/\theta X\}_m = (\alpha^? \Delta', \perp)$. L'hypothèse (2) prévoit de plus trois cas :
 - (a) Cas $\Delta' = \Delta$. Alors $\alpha^? \Delta' = \alpha^? \Delta$, et la proposition (2.a) est vérifiée.
 - (b) Cas $\Delta \sqsubseteq \Delta'$, $m = d$, $\alpha^? = \alpha$, et $[\Omega] \in \Delta'$. Alors $\alpha \Delta \sqsubseteq \alpha \Delta'$ et $[\Omega] \in \alpha \Delta'$, donc la proposition (2.b) est vérifiée.
 - (c) Cas $\Delta \sqsubseteq \Delta'$, $m = t$, $\alpha^? = \cdot$, et $[\Omega] \in |Z'| \cdot \Delta'$. Alors $\Delta \sqsubseteq \Delta'$ et $[\Omega] \in |\alpha : Z'| \cdot \Delta'$, donc la proposition (2.b) est vérifiée.
 - Cas $\{Z/\theta X\}_m = (\Delta, \sigma)$. Alors $\{\alpha : Z/\theta X\}_m = (\alpha^? \Delta, \sigma)$, et nous vérifions la proposition (3) sur $\{\alpha : Z/\theta X\}_m$. Par hypothèse (3) sur $\{Z/\theta X\}_m$ nous avons $\{Z'/\theta X\}_m = (\Delta, \sigma')$, donc $\{\alpha : Z'/\theta X\}_m = (\alpha^? \Delta, \sigma')$, avec $\alpha^? \Delta = \alpha^? \Delta$. L'hypothèse (3) donne de plus $\sigma \rightarrow \sigma'$, donc la proposition (3) est vérifiée.
- ii) Cas $\alpha : Z \rightarrow Y'$ par un pas de réduction à la racine. En particulier $\alpha : Z$ est un pré-radical, donc $\{\alpha : Z/\theta X\}_m = \bar{X}$. Raisonnons par cas sur $\{Y'/\theta X\}_m$.
- Cas $\{Y'/\theta X\}_m = \bar{X}$. Il n'y a rien à vérifier.
 - Cas $\{Y'/\theta X\}_m = (\Delta', \mu)$. Nous vérifions la proposition (1) sur $\{Y'/\theta X\}_m$. Par définition de la réduction étiquetée, Y' a la forme $[\Omega] : Z'$. Regardons $m \in \{d, t\}$.
 - * Cas $m = d$. Alors $\{[\Omega] : Z'/\theta X\}_d = [\Omega] : \{Z'/\theta X\}_d$. En particulier il existe une séquence Δ'' telle que $\Delta' = [\Omega] \Delta''$, donc $[\Omega] \in \Delta'$ et la proposition (1.a) est vérifiée.
 - * Cas $m = t$. Alors $|Y'| = |[\Omega] : Z'| = [\Omega] \cdot |Z'|$. En particulier $[\Omega] \in |Y'|$, donc $[\Omega] \in |Y'| \cdot \Delta'$ et la proposition (1.b) est vérifiée.
- V) Cas $\{\hat{x}/\theta \hat{x}\}_m = (\varepsilon, \{\})$ avec $x \in \theta$. Ce n'est par notre cas car l'argument est en forme normale.

VI) Cas $\{\{A_1A_2/\theta P_1P_2\}_m = \Gamma \cdot \{\{A_1/\theta P_1\}_t \uplus \{\{A_2/\theta P_2\}_d\}$ avec A_1A_2 et P_1P_2 des formes testables et Γ une séquence d'étiquettes éventuellement vide (Γ est $|P_1| \cdot |A_1|$ si $m = d$ et ε si $m = t$). Supposons $Y = A_1A_2 \rightarrow A'_1A_2 = Y'$ avec $A_1 \rightarrow A'_1$. En particulier il existe une séquence Γ' éventuellement vide telle que $\{\{A'_1A_2/\theta P_1P_2\}_m = \Gamma' \cdot \{\{A'_1/\theta P_1\}_t \uplus \{\{A_2/\theta P_2\}_d\}$ et nous pouvons vérifier que $\Gamma = \Gamma'$, par cas sur $m \in \{d, t\}$:

* Si $m = d$ alors $\Gamma = |P_1| \cdot |A_1|$ et $\Gamma' = |P_1| \cdot |A'_1|$. Or $|A'_1| = |A_1|$ par lemme 6.63, donc $\Gamma = \Gamma'$.

* Si $m = t$ alors $\Gamma = \varepsilon = \Gamma'$.

Comme $\{\{A_1A_2/\theta P_1P_2\}_m$ fait intervenir $\{\{A_1/\theta P_1\}_t$ et $\{\{A_2/\theta P_2\}_d$ et comme $\{\{A'_1A_2/\theta P_1P_2\}_m$ fait intervenir $\{\{A'_1/\theta P_1\}_t$ et $\{\{A_2/\theta P_2\}_d$, nous allons maintenant devoir considérer tous les cas pertinents sur les trois filtrages composés étiquetés $\{\{A_1/\theta P_1\}_t$, $\{\{A_2/\theta P_2\}_d$, et $\{\{A'_1/\theta P_1\}_t$. Raisonnons d'abord par cas sur la paire $\langle \{\{A_1/\theta P_1\}_t \mid \{\{A_2/\theta P_2\}_d \rangle$. Nous allons parcourir les neuf cases de la table de définition de \uplus .

- i) Cas $\langle \bar{X} \mid \bar{X} \rangle$. Alors $\{\{A_1A_2/\theta P_1P_2\}_m = \bar{X}$. Raisonnons par cas sur $\{\{A'_1/\theta P_1\}_t$.
- Cas $\{\{A'_1/\theta P_1\}_t = \bar{X}$. Alors $\{\{A'_1A_2/\theta P_1P_2\}_m = \bar{X}$, et il n'y a rien à vérifier.
 - Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \perp)$. Alors

$$\begin{aligned} \{\{A'_1A_2/\theta P_1P_2\}_m &= \Gamma \cdot (\{\{A'_1/\theta P_1\}_t \uplus \{\{A_2/\theta P_2\}_d\}) \\ &= \Gamma \cdot ((\Delta'_1, \perp) \uplus \bar{X}) \\ &= (\Gamma\Delta'_1, \perp) \end{aligned}$$

et nous vérifions la proposition (1) sur $\{\{A'_1A_2/\theta P_1P_2\}_m$. Regardons $m \in \{d, t\}$:

- * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1$. Donc $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1 = \Gamma\Delta'_1$ et la proposition (1.a) est vérifiée.
- * Cas $m = t$. Alors $\Gamma = \varepsilon$ et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1 = |A'_1A_2| \cdot \Gamma\Delta'_1$, donc la proposition (1.b) est vérifiée.
- Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \sigma'_1)$. Alors $\{\{A'_1A_2/\theta P_1P_2\}_m = \bar{X}$, et il n'y a rien à vérifier.

- ii) Cas $\langle \bar{X} \mid \Delta_2, \perp \rangle$. Alors $\{\{A_1A_2/\theta P_1P_2\}_m = (\Gamma\Delta_2, \perp)$, nous vérifions la proposition (2) sur $\{\{A'_1A_2/\theta P_1P_2\}_m$. Raisonnons par cas sur $\{\{A'_1/\theta P_1\}_t$.
- Cas $\{\{A'_1/\theta P_1\}_t = \bar{X}$. Alors $\{\{A'_1A_2/\theta P_1P_2\}_m = (\Gamma\Delta_2, \perp)$, avec $\Gamma\Delta_2 = \Gamma\Delta_2$, donc la proposition (2.a) est vérifiée.
 - Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \perp)$. Alors $\{\{A'_1A_2/\theta P_1P_2\}_m = (\Gamma\Delta'_1\Delta_2, \perp)$, avec $\Gamma\Delta_2 \sqsubseteq \Gamma\Delta'_1\Delta_2$. Regardons $m \in \{d, t\}$.

- * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1$. Donc $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1\Delta_2 = \Gamma\Delta'_1\Delta_2$ et la proposition (2.b) est vérifiée.
- * Cas $m = t$. $\Gamma = \varepsilon$ et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1$, donc $[\Omega] \in |A'_1| \cdot \Delta'_1\Delta_2 = |A'_1A_2| \cdot \Gamma\Delta'_1\Delta_2$ et la proposition (2.c) est vérifiée.

- Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \sigma'_1)\}$. Identique au cas $\{\{A'_1/\theta P_1\}_t = \bar{X}\}$.
- iii) Cas $\langle \bar{X} | \Delta_2, \sigma_2 \rangle$. Alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m = \bar{X}\}$. Raisonnons par cas sur $\{\{A'_1/\theta P_1\}_t\}$.
 - Cas $\{\{A'_1/\theta P_1\}_t = \bar{X}\}$. Alors $\{\{A'_1 A_2 / \theta P_1 P_2\}_m = \bar{X}\}$, et il n'y a rien à vérifier.
 - Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \perp)\}$. Alors $\{\{A'_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta'_1, \perp)\}$, et nous vérifions la proposition (1) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Regardons $m \in \{d, t\}$:
 - * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1$ donc $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1 = \Gamma \Delta'_1$ et la proposition (1.a) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$, et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1 = |A'_1 A_2| \cdot \Gamma \Delta'_1$, donc la proposition (1.b) est vérifiée.
 - Cas $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \sigma'_1)\}$. Alors $\{\{A'_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta'_1 \Delta_2, \sigma'_1 \uplus \sigma_2)\}$, et indépendamment de la nature de $\sigma'_1 \uplus \sigma_2$ (substitution σ' ou erreur \perp) nous vérifions la proposition (1) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Regardons $m \in \{d, t\}$:
 - * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1$, donc $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1 \Delta_2 = \Gamma \Delta'_1 \Delta_2$ et la proposition (1.a) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$ et par hypothèse (1.b) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $[\Omega] \in |A'_1| \cdot \Delta'_1 = |A'_1 A_2| \cdot \Gamma \Delta'_1$, donc la proposition (1.b) est vérifiée.
- iv) Cas $\langle \Delta_1, \perp | \bar{X} \rangle$. $\{\{A_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta_1, \perp)\}$, nous vérifions la proposition (2) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Par hypothèse (2) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \perp)\}$, donc $\{\{A'_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta'_1, \perp)\}$. L'hypothèse (2) prévoit de plus trois cas, dont deux s'appliquent ici (car nous appliquons l'hypothèse (2) sur $\{\{A'_1/\theta P_1\}_t\}$ qui est en mode t) :
 - (a) Cas $\Delta'_1 = \Delta_1$. Alors $\Gamma \Delta'_1 = \Gamma \Delta_1$ et la proposition (2.a) est vérifiée.
 - (c) Cas $\Delta_1 \sqsubseteq \Delta'_1$ et $[\Omega] \in |A'_1| \cdot \Delta'_1$. Alors $\Gamma \Delta_1 \sqsubseteq \Gamma \Delta'_1$ et nous regardons $m \in \{d, t\}$.
 - * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1 = \Gamma \Delta'_1$, donc la proposition (2.b) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$ et $[\Omega] \in |A'_1| \cdot \Delta'_1 = |A'_1 A_2| \cdot \Gamma \Delta'_1$, donc la proposition (2.c) est vérifiée.
- v) Cas $\langle \Delta_1, \perp | \Delta_2, \perp \rangle$. Alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta_1 \Delta_2, \perp)\}$, nous vérifions la proposition (2) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Par hypothèse (2) sur $\{\{A'_1/\theta P_1\}_t\}$ nous avons $\{\{A'_1/\theta P_1\}_t = (\Delta'_1, \perp)\}$, donc $\{\{A'_1 A_2 / \theta P_1 P_2\}_m = (\Gamma \Delta'_1 \Delta, \perp)\}$. L'hypothèse (2) prévoit de plus trois cas, dont seulement deux s'appliquent ici (car $\{\{A'_1/\theta P_1\}_t\}$ est en mode t) :
 - (a) Cas $\Delta'_1 = \Delta_1$. En particulier $\Gamma \Delta'_1 \Delta_2 = \Gamma \Delta_1 \Delta_2$, donc la proposition (2.a) est vérifiée.

- (c) Cas $\Delta_1 \sqsubseteq \Delta'_1$ et $[\Omega] \in |A'_1| \cdot \Delta'_1$. Alors $\Gamma \Delta_1 \Delta_2 \sqsubseteq \Gamma \Delta'_1 \Delta_2$ et nous regardons $m \in \{d, t\}$.
- * Cas $m = d$. Alors $\Gamma = |P_1| \cdot |A'_1|$, et $[\Omega] \in |P_1| \cdot |A'_1| \cdot \Delta'_1 \Delta_2 = \Gamma \Delta'_1 \Delta_2$, donc la proposition (2.b) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$ et $[\Omega] \in |A'_1| \cdot \Delta'_1 \Delta_2 = |A'_1 A_2| \cdot \Gamma \Delta'_1 \Delta_2$, donc la proposition (2.c) est vérifiée.
- vi) Cas $\langle \Delta_1, \perp | \Delta_2, \sigma_2 \rangle$. Identique au cas $\langle \Delta_1, \perp | \bar{X} \rangle$.
- vii) Cas $\langle \Delta_1, \sigma_1 | \bar{X} \rangle$. Alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m\} = \bar{X}$. De plus, par hypothèse (3) sur $\{\{A'_1 / \theta P_1\}_t\}$ nous avons $\{\{A'_1 / \theta P_1\}_t\} = (\Delta_1, \sigma'_1)$, donc $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\} = \bar{X}$, et il n'y a rien à vérifier.
- viii) Cas $\langle \Delta_1, \sigma_1 | \Delta_2, \perp \rangle$. Alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m\} = (\Gamma \Delta_2, \perp)$, nous vérifions la proposition (2) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Par hypothèse (3) sur $\{\{A'_1 / \theta P_1\}_t\}$ nous avons $\{\{A'_1 / \theta P_1\}_t\} = (\Delta_1, \sigma'_1)$, donc $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\} = (\Gamma \Delta_2, \perp)$. En particulier $\Gamma \Delta_2 = \Gamma \Delta_2$, donc la proposition (2.a) est vérifiée.
- ix) Cas $\langle \Delta_1, \sigma_1 | \Delta_2, \sigma_2 \rangle$. Alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m\} = (\Gamma \Delta_1 \Delta_2, \sigma_1 \uplus \sigma_2)$. Par hypothèse (3) sur $\{\{A'_1 / \theta P_1\}_t\}$ nous avons $\{\{A'_1 / \theta P_1\}_t\} = (\Delta_1, \sigma'_1)$ avec $\sigma_1 \rightarrow \sigma'_1$. Nous raisonnons sur les domaines de σ_1 et σ_2 .
- Si $\text{dom}(\sigma_1) \cap \text{dom}(\sigma_2) \neq \emptyset$ alors $\{\{A_1 A_2 / \theta P_1 P_2\}_m\} = (\Gamma \Delta_1 \Delta_2, \perp)$, nous vérifions la proposition (2) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Comme $\sigma_1 \rightarrow \sigma'_1$ nous avons de plus $\text{dom}(\sigma'_1) = \text{dom}(\sigma_1)$ et donc $\text{dom}(\sigma'_1) \cap \text{dom}(\sigma_2) \neq \emptyset$. Alors

$$\begin{aligned} \{\{A'_1 A_2 / \theta P_1 P_2\}_m\} &= (\Gamma \Delta_1 \Delta_2, \sigma'_1 \uplus \sigma_2) \\ &= (\Gamma \Delta_1 \Delta_2, \perp) \end{aligned}$$
 avec $\Gamma \Delta_1 \Delta_2 = \Gamma \Delta_1 \Delta_2$, et la proposition (2.a) est vérifiée.
 - Si $\text{dom}(\sigma_1) \cap \text{dom}(\sigma_2) = \emptyset$ alors il existe une substitution $\sigma = \sigma_1 \uplus \sigma_2$ telle que $\{\{A_1 A_2 / \theta P_1 P_2\}_m\} = (\Gamma \Delta_1 \Delta_2, \sigma)$, et nous vérifions la proposition (3) sur $\{\{A'_1 A_2 / \theta P_1 P_2\}_m\}$. Comme $\sigma_1 \rightarrow \sigma'_1$ nous avons de plus $\text{dom}(\sigma'_1) = \text{dom}(\sigma_1)$, et donc $\text{dom}(\sigma'_1) \cap \text{dom}(\sigma_2) = \emptyset$. Alors il existe une substitution $\sigma' = \sigma'_1 \uplus \sigma_2$ telle que $\{\{A'_1 A_2 / \theta P_1 P_2\}_t\} = (\Gamma \Delta_1 \Delta_2, \sigma')$, avec $\sigma = \sigma_1 \uplus \sigma_2 \rightarrow \sigma'_1 \uplus \sigma_2 = \sigma'$ et $\Gamma \Delta_1 \Delta_2 = \Gamma \Delta_1 \Delta_2$, donc la proposition (3) est vérifiée.

Le cas $Y = A_1 A_2 \rightarrow A_1 A'_2 = Y'$ avec $A_2 \rightarrow A'_2$ est similaire.

- VII) Cas $\{\{Y / \theta X\}_m\} = (\Gamma, \perp)$ avec Y, X formes testables, sinon, et Γ une séquence d'étiquettes éventuellement vide. Nous vérifions la proposition (2) sur $\{\{Y' / \theta X\}_m\}$. Deux cas sont possibles pour $\{\{Y / \theta X\}_m\} = (\Gamma, \perp)$.
- Cas où $X = \hat{x}$ avec $x \in \theta$ et $Y \neq \hat{x}$. Supposons $Y' = \hat{x}$. Alors $Y \rightarrow Y' = \hat{x}$ par un pas de réduction à la racine. Donc Y est un pré-radical, ce qui contredit le fait que Y est une forme testable. Donc $Y' \neq \hat{x}$ et il existe une séquence éventuellement vide Γ' telle que $\{\{Y / \theta X\}_m\} = (\Gamma', \perp)$. Regardons $m \in \{d, t\}$.
 - * Cas $m = d$. Alors $\Gamma = |Y|$ et $\Gamma' = |Y'|$. Par lemme 6.63 $|Y'| = |Y|$, donc $\Gamma = \Gamma'$ et la proposition (2.a) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$ et $\Gamma' = \varepsilon$. Donc $\Gamma = \Gamma'$ et la proposition (2.a) est vérifiée.

- Cas où $X = \langle \tau \rangle P.B.$. Alors il existe une séquence éventuellement vide Γ' telle que $\llbracket Y/\theta X \rrbracket_m = (\Gamma', \perp)$. Regardons $m \in \{d, t\}$.
 - * Cas $m = d$. Alors $\Gamma = |Y|$ et $\Gamma' = |Y'|$. Par lemme 6.63 $|Y'| = |Y|$, donc $\Gamma = \Gamma'$ et la proposition (2.a) est vérifiée.
 - * Cas $m = t$. Alors $\Gamma = \varepsilon$ et $\Gamma' = \varepsilon$. Donc $\Gamma = \Gamma'$ et la proposition (2.a) est vérifiée.
- VIII) Cas $\llbracket Y/\theta X \rrbracket_m = \bar{X}$, sinon. Deux cas sont possibles.
 - Cas où X est un terme étiqueté incomplet qui n'est pas une forme testable. Alors nous avons encore $\llbracket Y'/\theta X \rrbracket_m = \bar{X}$ et il n'y a rien à vérifier.
 - Cas où X est un terme étiqueté incomplet qui est une structure de données, et Y est un terme étiqueté incomplet qui n'est pas une forme testable. Regardons alors Y' :
 - Si Y' n'est pas une forme testable, alors nous avons encore $\llbracket Y'/\theta X \rrbracket_m = \bar{X}$ et il n'y a rien à vérifier.
 - Si Y' est une forme testable, alors $[\Omega] \in |Y'|$ par lemme 6.62. De plus $\llbracket Y'/\theta X \rrbracket_m = (\Delta, \mu)$, et nous vérifions la proposition (1) sur $\llbracket Y'/\theta X \rrbracket_m$. Regardons $m \in \{d, t\}$.
 - * Cas $m = d$. Alors $|Y'| \sqsubseteq \Delta$ par lemme 6.44, donc $[\Omega] \in \Delta$ et la proposition (1.a) est vérifiée.
 - * Cas $m = t$. Alors $[\Omega] \in |Y'| \cdot \Delta$, donc la proposition (1.b) est vérifiée. \square

Nous pouvons prouver de même à propos de la réduction dans le motif :

Lemme 6.67 (Filtrage composé étiqueté). *Soient X et Y deux termes étiquetés, $\rho : X \rightarrow X'$ un pas de réduction de nom Ω , et $m \in \{d, t\}$. Les deux propositions suivantes sont vérifiées :*

- (1) Si $\llbracket Y/\theta X \rrbracket_m = \bar{X}$ et $\llbracket Y/\theta X' \rrbracket_m = (\Delta', \mu')$, alors :
 - (a) Soit $m = d$ et $[\Omega] \in \Delta'$.
 - (b) Soit $m = t$ et $[\Omega] \in |X'| \cdot \Delta'$.
- (2) Si $\llbracket Y/\theta X \rrbracket_m = (\Delta, \perp)$, alors $\llbracket Y/\theta X' \rrbracket_m = (\Delta', \perp)$, avec trois cas :
 - (a) Soit $\Delta = \Delta'$.
 - (b) Soit $\Delta \sqsubseteq \Delta'$, $m = d$, et $[\Omega] \in \Delta'$.
 - (c) Soit $\Delta \sqsubseteq \Delta'$, $m = t$, et $[\Omega] \in |X'| \cdot \Delta'$.

Remarquons qu'il n'y a pas ici de cas sur $\llbracket Y/\theta X \rrbracket_m = (\Delta, \sigma)$. Ceci vient d'un fait général du *Pure Pattern Calculus* [JK09] selon lequel le motif est alors une forme normale.

Remarque 6.68. *Le lemme 6.65 concerne les effets d'une réduction dans l'argument d'un filtrage. Les phénomènes qu'il décrit sont donc indépendants du polymorphisme de motif du Pure Pattern Calculus.*

Nous pouvons maintenant désentrelacer les contenus des lemmes 6.65 et 6.67 sur le filtrage composé étiqueté pour obtenir un résultat sur la réduction dans les

termes d'un filtrage non-encore défini (lemme 6.69), un résultat sur la réduction dans les termes d'un filtrage déjà défini et qui échoue (lemme 6.70), et un résultat sur la réduction dans les termes d'un filtrage déjà défini et qui réussit (lemme 6.71). Le premier sera la clé du lemme 6.59 de contribution directe et les derniers ouvriront le lemme 6.60 de stabilité des radicaux.

Lemme 6.69 (Réduction dans un filtrage indéfini). *Soient X et Y deux termes étiquetés tels que $\{Y/\theta X\}$ n'est pas défini. Alors les deux propositions suivantes sont vérifiées :*

- Si $Y \rightarrow Y'$ est un pas de réduction de nom Ω et si $\{Y'/\theta X\} = (\Delta', \mu')$, alors $[\Omega] \in \Delta'$.
- Si $X \rightarrow X'$ est un pas de réduction de nom Ω et si $\{Y/\theta X'\} = (\Delta', \mu')$, alors $[\Omega] \in \Delta'$.

Démonstration. Par définition, le filtrage $\{Y/\theta X\}$ n'est pas défini si et seulement si $\{\{Y/\theta X\}_d\} = \bar{X}$. Supposons que $Y \rightarrow Y'$ est un pas de réduction de nom Ω et que $\{Y'/\theta X\} = (\Delta', \mu')$. Alors il existe μ'' tel que $\{\{Y'/\theta X\}_d\} = (\Delta', \mu'')$. Donc le lemme 6.65, proposition (1.a), s'applique sur $\{\{Y'/\theta X\}_d\}$, et $[\Omega] \in \Delta'$.

De même avec le lemme 6.67, proposition (1), si $X \rightarrow X'$ est un pas de réduction de nom Ω et $\{Y/\theta X'\} = (\Delta', \mu')$. \square

Nous allons avec ce lemme 6.69 pouvoir démontrer le lemme 6.59 de contribution directe, dont l'énoncé est maintenant rappelé :

Lemme 6.59. *Pour tout pas de réduction $\rho : T \rightarrow T'$ de nom Ω , si ρ_c est un pas de réduction de source T' et de nom Ω_c qui est créé par ρ , alors $\Omega \hookrightarrow^d \Omega_c$.*

Démonstration. Par cas sur les créations de radicaux, avec l'extension immédiate du lemme 6.29 au cadre étiqueté.

- (1) Le radical créé $\text{dex}(\rho_c)$ est de la forme $\alpha_c : ((\Gamma_c : \text{duct}(\rho))A_c)$, où le réduit $\text{duct}(\rho)$ est de la forme $[\Omega] : Z'$. Donc le nom Ω_c admet pour préfixe la séquence $\alpha_c \Gamma_c [\Omega]$, et en particulier $\Omega_c \hookrightarrow^d \Omega$.
- (2⁺) Le radical $\text{duct}(\rho)$ est de la forme $\alpha : ((\Gamma : \langle \theta \rangle P.S[B_1, \dots, B_n])A)$ où S est le θ -squelette de $B = S[B_1, \dots, B_n]$, le réduit $\text{duct}(\rho)$ est de la forme $[\Omega] : [\Omega, S]^\sigma [B_1, \dots, B_n]$, et la position de la racine du radical créé $\text{dex}(\rho_c)$ descend d'une position du θ -squelette de $B = S[B_1, \dots, B_n]$, c'est-à-dire d'une position de S . Donc la position racine de $\text{dex}(\rho_c)$ est dans $[\Omega, S]$, et le radical $\text{dex}(\rho_c)$ est de la forme $[\Omega, \alpha] : (T_1 T_2)$. En particulier $[\Omega, \alpha]$ est un préfixe de Ω_c , et $\Omega \hookrightarrow^d \Omega_c$.
- (3,4) Notons le radical créé $\text{dex}(\rho_c) = \alpha_c : ((\Gamma_c \langle \theta_c \rangle P_c.B_c)A_c)$, et $\{A_c/\theta_c P_c\} = (\Delta_c, \mu_c)$. Le nom du pas de réduction ρ_c est $\alpha_c \Gamma_c \Delta_c$. L'ancêtre $\alpha_c : ((\Gamma_c \langle \theta_c \rangle P_a.B_c)A_a)$ du radical créé $\text{dex}(\rho_c)$ est tel que $\{A_a/\theta_c P_a\}$ n'est pas défini, et le pas de réduction ρ a lieu soit dans A_a soit dans P_a . Par lemme 6.69 nous avons $[\Omega] \in \Delta_c$, donc $\Omega \hookrightarrow^d \Omega_c$. \square

Lemme 6.70 (Réduction dans un filtrage défini qui échoue). *Soient X et Y deux termes étiquetés tels que $\{Y/\theta X\} = (\Delta, \perp)$. Alors les deux propositions suivantes sont vérifiées :*

- Si $Y \rightarrow Y'$ est un pas de réduction de nom Ω alors $\{Y'/\theta X\} = (\Delta', \perp)$ avec $\Delta = \Delta'$ ou $[\Omega] \in \Delta'$.
- Si $X \rightarrow X'$ est un pas de réduction de nom Ω alors $\{Y/\theta X'\} = (\Delta', \perp)$ avec $\Delta = \Delta'$ ou $[\Omega] \in \Delta'$.

Démonstration. Considérons un pas de réduction $Y \rightarrow Y'$. Par définition, il y a deux cas possibles pour $\{Y/\theta X\} = (\Delta, \perp)$:

- Cas $\{\!\{Y/\theta X\}\!\}_d = (\Delta, \perp)$. Alors par lemme 6.65, proposition (2), il existe une séquence Δ' telle que $\{Y'/\theta X\} = (\Delta', \perp)$, et l'une des propositions (2.a) ou (2.b) est vérifiée, soit $\Delta = \Delta'$ ou $[\Omega] \in \Delta'$ (la proposition (2.c) n'est pas envisageable en mode d).
- Cas $\{\!\{Y/\theta X\}\!\}_d = (\Delta, \sigma)$ avec $\text{dom}(\sigma) \neq \theta$. Alors par lemme 6.65, proposition (3), il existe une substitution σ' telle que $\sigma \rightarrow \sigma'$ et $\{\!\{Y'/\theta X\}\!\}_d = (\Delta, \sigma')$. Comme $\sigma \rightarrow \sigma'$, alors en particulier $\text{dom}(\sigma') = \text{dom}(\sigma)$ et $\text{dom}(\sigma') \neq \theta$. Donc $\{\!\{Y'/\theta X\}\!\}_d = (\Delta, \perp)$.

Considérons maintenant un pas de réduction $X \rightarrow X'$. Par définition, il y a deux cas possibles pour $\{Y/\theta X\} = (\Delta, \perp)$:

- Cas $\{\!\{Y/\theta X\}\!\}_d = (\Delta, \perp)$. Nous concluons de la même manière que pour la réduction de Y , avec le lemme 6.67.
- Cas $\{\!\{Y/\theta X\}\!\}_d = (\Delta, \sigma)$ avec $\text{dom}(\sigma) \neq \theta$. Alors par propriété de PPC [JK09], X est une forme normale, ce qui n'est pas notre cas. \square

Lemme 6.71 (Réduction dans un filtrage défini qui réussit). *Soient X et Y deux termes étiquetés tels que $\{Y/\theta X\} = (\Delta, \sigma)$. Pour tout pas de réduction $Y \rightarrow Y'$ il existe une substitution σ' telle que $\sigma \rightarrow \sigma'$ et $\{Y'/\theta X\} = (\Delta, \sigma')$.*

Démonstration. Par définition de $\{Y/\theta X\} = (\Delta, \sigma)$, nous avons $\{\!\{Y/\theta X\}\!\}_d = (\Delta, \sigma)$ avec $\text{dom}(\sigma) = \theta$. Alors par lemme 6.65, proposition (3), il existe une substitution σ' telle que $\sigma \rightarrow \sigma'$ et $\{\!\{Y'/\theta X\}\!\}_d = (\Delta, \sigma')$. Comme $\sigma \rightarrow \sigma'$, alors en particulier $\text{dom}(\sigma') = \text{dom}(\sigma) = \theta$. Donc $\{Y'/\theta X\} = (\Delta, \sigma')$. \square

À l'aide de ces lemmes 6.70 et 6.71, nous pouvons démontrer le lemme 6.60 de stabilité des radicaux, dont l'énoncé est maintenant rappelé :

Lemme 6.60. *Soit $\rho : X \rightarrow X'$ un pas de réduction de nom Ω . Soit ρ_a un pas de réduction de source X et de nom Ω_a et de radical R_a . Si R_d est un descendant de R_a dans X' , alors R_d est le radical d'un pas de réduction ρ_d (résidu de ρ_a) et de nom Ω_d tel que :*

- Si $\Omega_a \neq \Omega_d$ alors $\Omega \hookrightarrow^d \Omega_d$.
- Pour tout nom Ω' tel que $\Omega' \hookrightarrow \Omega_a$ nous avons aussi $\Omega' \hookrightarrow \Omega_d$.

Démonstration. Par cas sur les positions relatives de ρ_a et ρ , avec les lemmes 6.70 et 6.71. \square

Ainsi, le lemme 6.59 de contribution directe nous assure que dans tout cas où un pas de réduction ρ crée un pas de réduction ρ_c le nom Ω de ρ apparaît dans le nom Ω_c de ρ_c . Le lemme 6.60 de stabilité assure ensuite que le nom Ω'_c de tout

résidu ρ'_c de ρ_c contient encore le contributeur Ω . En particulier, nous pouvons donc espérer que toute information relative à la création d'un pas de réduction sera toujours contenue dans son nom, ce qui correspondrait à la complétude causale. En revanche, rien ne permet de dire à ce stade que les étiquettes ne contiennent pas de « faux positifs » quant aux informations de dépendance, et il y a donc un danger potentiel pour la correction causale.

Enfin, par dessus ces deux remarques nous allons voir à la section 6.7 suivante que le *Pure Pattern Calculus* demande de reconsidérer la notion de nécessité. Ceci relativise dans ce cadre la légitimité des énoncés de correction et de complétude causales, qui sont basés sur cette nécessité.

6.7 Discussion : non-stabilité

Dans ce chapitre nous avons construit un étiquetage pour le *Pure Pattern Calculus* avec un objectif particulier : capturer dans les étiquettes toutes les informations relatives aux dépendances causales entre les différents pas de réduction. Dans cette optique chaque pas de réduction (identifié à son radical) est muni d'un nom construit à partir de ses étiquettes, et une relation d'ordre sur les noms vise à rendre compte des dépendances causales entre les pas de réduction.

Aux origines de cette approche [Lé80], les noms fournissaient une identification définitive des radicaux. Ainsi, à sa création chaque radical se voyait attribuer un nom, et ce nom était transmis *verbatim* à chaque résidu, garantissant une identification pérenne d'une certaine famille de radicaux. Cependant...

6.7.1 Instabilité des noms

Cette stabilité des radicaux ne tient pas dans le calcul que nous avons défini dans ce chapitre. En effet, comme le sous-entend l'énoncé du lemme 6.60 de stabilité des radicaux, il existe des cas dans lesquels le nom d'un radical peut changer.

Ces cas sont à comprendre comme ceux dans lesquels un pas de réduction ne détruit pas le radical observé mais apporte une modification significative à sa structure, comme l'illustre l'exemple 6.72.

Exemple 6.72 (Altération d'un radical).

Considérons le terme du *Pure Pattern Calculus*

$$(\langle \emptyset \rangle \hat{c} \hat{d}.b)((I\hat{d})\hat{c})$$

où I désigne l'identité $\langle x \rangle \hat{x}.x$, et son étiquetage complet

$$\alpha : @(\beta : \langle \emptyset \rangle \gamma : @(\delta : \hat{c}, \epsilon : \hat{d}).B, \zeta : @(\eta : @(\theta : \langle x \rangle \iota : \hat{x}.x, \kappa : \hat{d}), \mu : \hat{c}))$$

Ce terme est un pré-radical, et son filtrage

$$\{(I\hat{d})\hat{c}/_{\emptyset} \hat{c}\hat{d}\} \\ \{\zeta : @(\eta : @(\theta : \langle x \rangle \iota : \hat{x}.x, \kappa : \hat{d}), \mu : \hat{c})/_{\emptyset} \gamma : @(\delta : \hat{c}, \epsilon : \hat{d})\}$$

se résoud en la paire

$$(\gamma\zeta\epsilon\mu, \perp)$$

Ainsi le terme d'origine est un radical de nom

$$\alpha\beta\gamma\zeta\epsilon\mu$$

qui se réduit en

$$\lceil \alpha\beta\gamma\zeta\epsilon\mu \rceil : T_{\perp}$$

Mais notre terme contient également un radical interne

$$\eta : @(\theta : \langle x \rangle_l : \hat{x}.x, \kappa : \hat{d})$$

dont le réduit est

$$\lceil \eta \theta_l \rceil : \kappa : \hat{d}$$

Après ce pas de réduction, le terme cible

$$\begin{aligned} & (\langle \emptyset \rangle \hat{c} \hat{d}.b)(\hat{d} \hat{c}) \\ \alpha : @(\beta : \langle \emptyset \rangle \gamma : @(\delta : \hat{c}, \epsilon : \hat{d}).B, \zeta : @(\lceil \eta \theta_l \rceil : \kappa : \hat{d}, \mu : \hat{c})) \end{aligned}$$

est toujours un pré-radical. Son filtrage

$$\begin{aligned} & \{ \hat{d} \hat{c} /_{\emptyset} \hat{c} \hat{d} \} \\ \{ \zeta : @(\lceil \eta \theta_l \rceil : \kappa : \hat{d}, \mu : \hat{c}) /_{\emptyset} \gamma : @(\delta : \hat{c}, \epsilon : \hat{d}) \} \end{aligned}$$

est toujours défini, et se résoud en la paire

$$(\gamma\zeta\delta\lceil \eta \theta_l \rceil \kappa \epsilon \mu, \perp)$$

Alors, le second terme est bien un radical résidu du premier, mais son nom est maintenant

$$\alpha\beta\gamma\zeta\delta\lceil \eta \theta_l \rceil \kappa \epsilon \mu$$

et son réduit est

$$\lceil \alpha\beta\gamma\zeta\delta\lceil \eta \theta_l \rceil \kappa \epsilon \mu \rceil : \perp$$

Bilan : le nom d'un radical peut changer, impliquant au passage un défaut de confluence du calcul. Ce phénomène est lié à la manière dont les erreurs de filtrage sont détectées dans PPC. L'exemple 6.72 précédent concerne en effet un terme

$$(\langle \emptyset \rangle \hat{c} \hat{d}.b) ((I\hat{d}) \hat{c})$$

qui est un radical dont le filtrage $\{(I\hat{d}) \hat{c} /_{\emptyset} \hat{c} \hat{d}\}$ échoue par la comparaison entre le \hat{d} du motif et le \hat{c} de l'argument. Dans ce cas la comparaison entre le \hat{c} du motif et le $I\hat{d}$ de l'argument est indéfinie (\bar{X}) et donc ignorée. Après réduction nous obtenons un nouveau radical

$$(\langle \emptyset \rangle \hat{c} \hat{d}.b) (\hat{d} \hat{c})$$

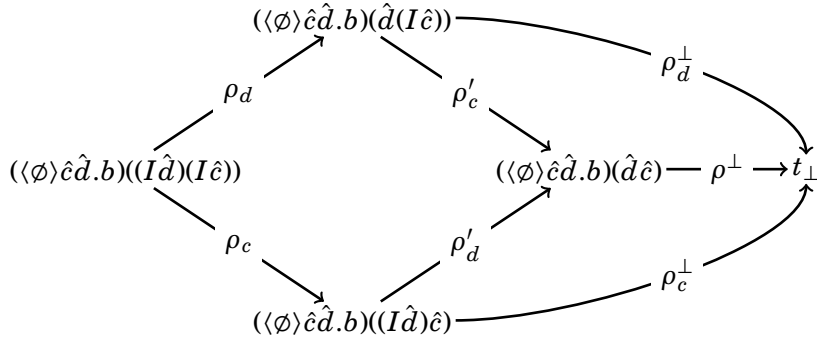
dont le filtrage $\{\hat{d}\hat{c}/_{\emptyset}\hat{c}\hat{d}\}$ a maintenant deux bonnes raisons d'échouer, les branches gauche et droite étant toutes deux définies.

Ainsi, dans le premier cas les étiquettes ne témoignaient que de l'erreur de filtrage déjà visible, mais dans le deuxième cas les étiquettes font apparaître les deux erreurs. En effet, dans ce dernier cas rien dans le terme ne justifie de privilégier une erreur par rapport à l'autre. Elles apparaissent toutes deux également responsables de l'échec de filtrage.

Ceci incite à une nouvelle réflexion sur la notion de nécessité.

6.7.2 Non-stabilité et nécessité

Nous avons dans le *Pure Pattern Calculus* l'espace de réduction suivant :



Dans cet espace de réduction, ρ^\perp est résidu à la fois de ρ_c^\perp et de ρ_d^\perp , et ainsi les trois pas de réduction ρ^\perp , ρ_c^\perp , et ρ_d^\perp représentent le même « événement » de réécriture. En d'autres termes, la « création » de ρ^\perp correspond à la création de ρ_c^\perp ou de ρ_d^\perp . Or ρ_c^\perp est créé par ρ_c et ρ_d^\perp est créé par ρ_d , et les deux pas de réduction ρ_c et ρ_d sont indépendants l'un de l'autre. Alors chacun de ρ_c et ρ_d est suffisant pour la création de ρ^\perp .

Nous appellerons *non-stabilité* cette possibilité que deux pas de réduction indépendants ρ_c et ρ_d soient chacun une condition suffisante à la création d'un même pas de réduction ρ^\perp . Cette notion fait référence aux axiomes de stabilité énoncés par Glauert et Khasidashvili [GK96] et Melliès [Mel05] pour bannir ce genre de comportement qui remet en question la pertinence de la notion usuelle de nécessité.

En effet, dans notre système non-stable aucun des deux pas de réduction ρ_c ou ρ_d pris isolément n'est nécessaire à la création de ρ^\perp , puisque l'autre sera toujours suffisant. Autrement dit la notion de nécessité usuelle ne détecte aucun pas de réduction nécessaire. Cependant, ne rien faire ne suffit pas à la création de ρ^\perp , car il faut quand même pour cela que l'un au moins des pas de réduction de l'ensemble $\{\rho_c, \rho_d\}$ soit effectué.

Nous pourrions donc qualifier l'ensemble $\{\rho_c, \rho_d\}$ d'*ensemble nécessaire* pour la création de ρ^\perp , dans le sens qu'il est nécessaire qu'au moins un de ces deux pas de réduction soit effectué pour aboutir à la création de ρ^\perp .

Cette idée d'ensemble nécessaire a été introduite par Sekar et Ramakrishnan dans un cadre de réécriture du premier ordre [SR90], et portée au *Pure*

Pattern Calculus par Bonelli, Kesner, Lombardi et Ríos [BKLR12]. Dans les deux cas la « nécessité » était relative à l'obtention d'une forme normale, et la notion d'ensemble nécessaire avait pour but d'obtenir une stratégie de réduction normalisante dans un système non-séquentiel (c'est-à-dire un système dans lequel il n'existe pas toujours un pas de réduction nécessaire à la normalisation).

Dans la présente discussion nous changeons légèrement de point de vue, puisque la « nécessité » n'est pas relative à la normalisation mais à la création d'un pas de réduction, et car ce n'est pas contre la non-séquentialité mais contre la non-stabilité que nous luttons. Pour indiquer la différence entre ces deux notions, pointons l'exemple de Berry [Ber78] qui est un système orthogonal et stable mais non-séquentiel. Autrement dit, la non-séquentialité n'implique pas la non-stabilité. En revanche, comme nous le verrons à la section 6.7.3, dans un système qui ne serait ni stable ni séquentiel, forcer la séquentialité peut avoir pour effet de ramener la stabilité.

L'élaboration d'une théorie des dépendances causales entre pas de réduction qui soit basée non sur la nécessité usuelle mais sur la notion d'ensemble nécessaire semble être un défi susceptible d'ouvrir de nouveaux horizons. Cela signifierait quitter le domaine de la causalité « conjonctive » où nous répertorions pour chaque événement un ensemble nécessaire et suffisant d'antécédents pour le domaine de la causalité « disjonctive » où nous pouvons exprimer des alternatives entre plusieurs combinaisons indépendantes permettant d'aboutir au même événement. Et là, les formes que peuvent prendre des notions comme la standardisation ou la réduction optimale sont encore à découvrir.

6.7.3 Stabilisation

Retour sur Terre. Les théories de l'optimalité auxquelles nous allons nous raccrocher au chapitre 10 ne sont bien établies que dans le cadre de systèmes stables. Imaginons donc que nous voulons stabiliser l'identification des radicaux dans le *Pure Pattern Calculus* étiqueté, par exemple en vue de conserver la notion de nécessité usuelle et de poser les bases d'une théorie du partage optimal pour une version stabilisée de *PPC* (éventuellement en réduction faible, mais ce n'est pas le point critique ici). Quels moyens avons-nous d'annuler la non-stabilité du calcul ?

En reprenant l'exemple 6.72 précédent regardons le terme

$$(\langle \emptyset \rangle \hat{c} \hat{d}.B) (\hat{d} \hat{c})$$

et son filtrage $\{\hat{d} \hat{c} /_{\emptyset} \hat{c} \hat{d}\}$. Comment peut-on, parmi les deux erreurs, désigner un « responsable » de l'erreur globale ?

Une première solution consiste à désigner comme également responsables les deux erreurs. Pour marier cette discipline avec la stabilité, il faut soit que dès l'occurrence d'une première erreur nous soyons capables de détecter aussi la deuxième, soit que nous ne considérions pas le radical comme tel avant l'apparition des deux erreurs. Le premier de ces cas est des plus glissants, puisqu'il demande d'anticiper certains résultats des réductions futures, anticipation dont

nous pouvons attendre qu'elle soit indécidable. Le deuxième cas est en revanche tout à fait réalisable, et correspond à imposer l'exploration de l'intégralité des branches du motif lors de tout filtrage. Nous pouvons obtenir cet effet en considérant la définition alternative de l'opération d'union disjointe sur les filtrats utilisée dans la première version de *PPC* [JK06] et dans le livre de Jay [Jay09]. Voici ladite définition alternative, en incluant les étiquettes :

\uplus	(Δ_2, σ_2)	(Δ_2, \perp)	\bar{X}
(Δ_1, σ_1)	$(\Delta_1 \Delta_2, \sigma_1 \uplus \sigma_2)$	$(\Delta_1 \Delta_2, \perp)$	\bar{X}
(Δ_1, \perp)	$(\Delta_1 \Delta_2, \perp)$	$(\Delta_1 \Delta_2, \perp)$	\bar{X}^*
\bar{X}	\bar{X}	\bar{X}^*	\bar{X}

Les différences avec notre définition de référence du *Pure Pattern Calculus* [JK09] sont les deux cas symétriques marqués d'une étoile, concernant l'union d'un filtrage composé indéfini avec un filtrage composé qui échoue. Dans la définition de référence *PPC*, l'échec global est détecté même si certaines branches sont indéfinies. Dans la présente variante ce n'est pas le cas, il faut impérativement que toutes les branches soient définies pour que le filtrage global soit défini. Une conséquence de cette solution est donc de rejeter les deux filtrages $\{\hat{d}(I\hat{c})/\phi \hat{c}\hat{d}\}$ et $\{(I\hat{d})\hat{c}/\phi \hat{c}\hat{d}\}$ comme indéfinis. Cette discipline produit une seconde différence par rapport à l'étiquetage de la section 6.4, avec les deux cas symétriques « σ contre \perp ». À la section 6.4 nous ne conservions que les étiquettes relatives à la branche qui échouait (\perp), mais ici tout est collecté. En effet il est utile dans ce cas de savoir que la branche qui réussit (σ) est bien définie, pour différencier ce cas du cas « \bar{X} contre \perp ».

Une deuxième solution consiste à désigner arbitrairement un bouc émissaire, selon une stratégie déterministe prédéfinie. Dans cette voie, la non-stabilité est éliminée par suppression de la *non-séquentialité* du système. Autrement dit, nous imposons une stratégie de filtrage pour séquentialiser artificiellement les opérations de filtrage. Nous pouvons par exemple considérer que les branches droites sont toujours explorées les premières. Cette stratégie correspond à une nouvelle variante de l'union disjointe :

\uplus	(Δ_2, σ_2)	(Δ_2, \perp)	\bar{X}
(Δ_1, σ_1)	$(\Delta_1 \Delta_2, \sigma_1 \uplus \sigma_2)$	$(\Delta_2, \perp)^*$	\bar{X}
(Δ_1, \perp)	$(\Delta_1 \Delta_2, \perp)$	$(\Delta_2, \perp)^*$	\bar{X}^*
\bar{X}	\bar{X}	$(\Delta_2, \perp)^*$	\bar{X}

Cette version brise complètement la symétrie des opérations précédentes. Ici, dès que la branche de droite est indéfinie, tout le filtrage l'est également. De même, dès que la branche de droite renvoie une erreur de filtrage, alors le filtrage global échoue, et *seule l'histoire de la branche droite est transmise*. Enfin, le résultat et l'histoire de la branche gauche sont pris en compte seulement dans le cas où la branche droite est définie et réussit. Ceci a plusieurs conséquences. D'abord le filtrage $\{\hat{d}(I\hat{c})/\phi \hat{c}\hat{d}\}$ est indéfini car sa branche droite est indéfinie et le filtrage $\{(I\hat{d})\hat{c}/\phi \hat{c}\hat{d}\}$ est défini malgré la non-définition de sa branche gauche. Ensuite,

le radical de l'exemple 6.72 ne change pas de nom après réduction, puisque les seuls changements d'étiquettes concernent la branche gauche. Cette solution (mais avec la priorité à gauche) est celle qui a été développée dans la première année de cette thèse [Bal10a].

Ces deux solutions ont pour caractéristique commune de redonner une forme usuelle à la notion de nécessité, et ainsi de rendre pertinents les énoncés de correction et de complétude causales du chapitre 5. La validité de ces énoncés a été essentiellement vérifiée dans le cadre de la deuxième solution [Bal10b], et est une conjecture très tentante dans le cadre de la première solution.

Bilan du chapitre

Dans ce chapitre nous avons porté l'analyse de causalité faite pour le λ -calcul faible au chapitre 5 au cadre significativement plus complexe du *Pure Pattern Calculus* [JK09].

Dans ce calcul doté de mécanismes de filtrage riches, nous avons rencontré deux problèmes de fond. D'abord, de nouveaux cas de création de radicaux apparaissent (section 6.3), qui demandent d'enrichir le mécanisme de fabrication du nom d'un radical (section 6.4). Ensuite (section 6.7), la non-stabilité du système vient interférer avec la notion de nécessité qui permettait au chapitre 5 de valider l'interprétation causale des étiquettes.

Nous avons également pu observer un problème de forme, en ce que l'utilisation de règles de réduction complexes et définies au niveau *méta* rendaient plus difficiles à la fois l'analyse de la causalité et l'attribution de noms aux radicaux (sections 6.3 et 6.4). Le chapitre suivant va montrer comment l'utilisation d'un cadre plus explicite comme les *CRS* simplifie significativement la donne.

CHAPITRE 7

Étiquetage causal des *CRS* faibles

Sommaire

7.1 Causalité dans les <i>CRS</i>	221
7.2 Étiquetage	228
7.3 Équivalence avec le partage pleinement paresseux	234
7.4 Propriétés de contribution	241
7.5 Correction et complétude causales	244

Nous allons maintenant développer les idées des chapitres 5 et 6, et étudier dans le cadre des *CRS* faibles des étiquetages qui intègrent des informations sur les dépendances causales entre les différents pas de réduction. Cette construction est un élément clé du lien entre partage pleinement paresseux et partage optimal en réduction faible, puisque d'une part les résultats de causalité seront un élément essentiel du chapitre 10 d'optimalité, et d'autre part nous allons prouver que les étiquettes reflétant les dépendances causales définissent un *SLS* qui spécifie la même réduction partagée que l'étiquetage pleinement paresseux du chapitre 4.

La section 7.1 analyse les cas dans lesquels un pas de réduction peut créer un autre pas de réduction dans les *CRS* faibles, puis la section 7.2 convertit cette analyse en un étiquetage des *CRS* faibles. La section 7.3 montre que l'étiquetage obtenu définit un *SLS* dont la réduction parallèle étiquetée est bisimilaire à la réduction partagée pleinement paresseuse. La section 7.4 établit les propriétés causales de l'étiquetage de ce chapitre qui sont valides dans tout *CRS* faible, et enfin la section 7.5 se restreint aux *CRS* faibles orthogonaux pour démontrer dans ce cadre la correction et la complétude causales de notre étiquetage.

7.1 Causalité dans les *CRS*

Dans les chapitres 5 et 6 précédents nous avons cherché à décrire, dans les cas du λ -calcul faible et du *Pure Pattern Calculus* faible, les relations causales

entre les différents « événements de réduction ».

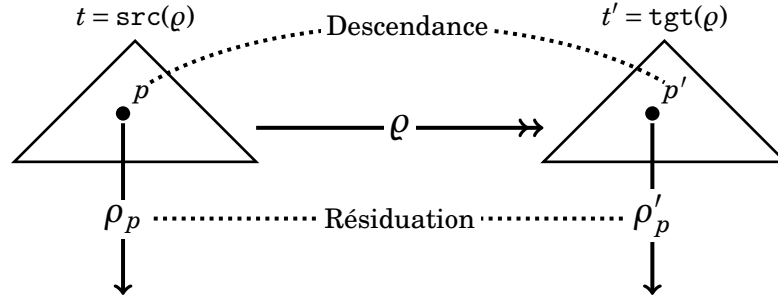
Après avoir décrit lesdits événements comme des familles de pas de réduction, nous nous sommes concentrés sur une notion de contribution, qui dénotait non pas les *contributions à la création d'un pas de réduction*, mais les *contributions à la création d'un radical*. Nous avons en fait profité du fait que dans les deux systèmes concernés les pas de réduction ayant pour source un certain terme t pouvaient être identifiés aux pré-radicaux non-gelés de t .

Dans le cadre général des *CRS*, et en particulier dans le cadre des *CRS* non-orthogonaux, ce glissement n'est plus anodin. Pour mener une analyse des contributions nous devons d'abord préciser ce qu'est la résiduation dans un système potentiellement non-orthogonal.

7.1.1 Retour sur la création et la résiduation

Deux concepts hautement corrélés mais néanmoins distincts sont couramment invoqués quand il s'agit de suivre l'évolution d'un terme le long d'une séquence de réduction :

- La *descendance*, qui trace le devenir des *positions* des termes.
- La *résiduation*, qui trace le devenir des *pas de réduction* ayant pour sources les termes considérés.



Les deux concepts ont d'ailleurs été déjà abondamment utilisés dans les précédents chapitres de cette thèse. Mais jusqu'ici, leurs différences fondamentales ont été ignorées. En particulier, les résidus (qui sont donc des pas de réduction) ont été définis dans le λ -calcul et le *Pure Pattern Calculus* comme des radicaux (c'est-à-dire des sous-termes) dont les positions vérifient certaines propriétés de descendance. Cette confusion relativement courante était possible car en λ -calcul comme dans *PPC* chaque radical définit exactement un pas de réduction, sans ambiguïté possible.

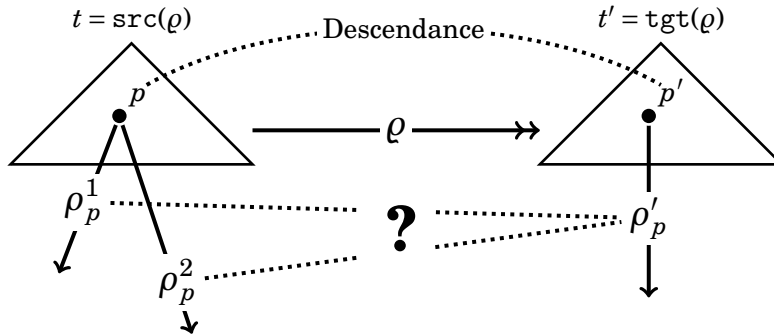
Ainsi dans les chapitres 5 et 6, si nous considérons une séquence de réduction $\rho : t \rightarrow t'$, des positions $p \in \text{pos}(t)$ et $p' \in \text{pos}(t')$ telles que p' descend de p , et des pas de réduction ρ_p et ρ'_p de sources respectives t et t' et de positions racines respectives p et p' , alors nous pouvions mener le raisonnement suivant : le sous-terme $t|_p$ est un radical qui définit ρ_p ; or le sous-terme $t'|_{p'}$ est « ce qui reste » de $t|_p$ après ρ , et de plus ce sous-terme $t'|_{p'}$ est un radical qui définit ρ'_p ; comme enfin les deux pas de réduction ρ_p et ρ'_p ont « même nature » (β -réduction

en λ -calcul, β_m -réduction dans PPC) nous pouvons légitimement considérer ρ'_p comme un résidu de ρ_p après ρ .

Cette confusion simplificatrice ne tient plus dans le cadre général des CRS, et notamment des CRS non-orthogonaux. En effet, dans un tel système nous pouvons observer une variété de nouveaux cas :

- Plusieurs pas de réduction possibles pour un même radical.
- Concurrément à la descendance, remplacement d'un pas de réduction par un autre qui ne lui est pas apparenté.

Il conviendra donc à partir de maintenant de ne pas oublier ce point : la relation de résiduation est une relation sur les pas de réduction, qui ne peut être déduite de la simple existence d'un radical. Dans le schéma ci-dessous par exemple, en supposant que ρ'_p ait au plus un ancêtre le long d'une séquence de réduction donnée, il faut forcément que l'un des deux candidats ρ_p^1 ou ρ_p^2 se « désiste ». Et à partir du moment où ces désistements sont possibles, rien n'assure que même un seul de ρ_p^1 ou ρ_p^2 soit un ancêtre de ρ'_p . En d'autres termes, les radicaux ne sont que ce que leur nom indique : des structures prêtes à réagir, et qui peuvent réagir éventuellement de plusieurs manières différentes¹. L'exemple 7.2 en donnera une illustration concrète.



Étant donné deux pas de réductions dont les radicaux sont en relation d'ancêtre et de descendant, nous distinguerons les cas de *résiduation exacte* où les deux pas viennent de la même règle de réécriture des cas de *fausse résiduation* où les deux pas sont liés à des règles différentes. Il sera également pertinent à la section 7.4 de distinguer un troisième cas de *résiduation approchée* où les deux pas utilisent des règles « apparentées ».

Définition 7.1 (Résidus). Soient un pas de réduction $\rho : t_1 \rightarrow t_2$, des positions $p_1 \in \text{pos}(t_2)$ et $p_2 \in \text{pos}(t_2)$ telles que $p_2 \in p_1/\rho$, et des pas de réduction ρ_1 et ρ_2 de sources respectives t_1 et t_2 et de positions racines respectives p_1 et p_2 . Comparons les règles $l_1 \rightarrow r_1$ et $l_2 \rightarrow r_2$ des pas de réduction respectifs ρ_1 et ρ_2 .

- Si les règles sont égales, alors le pas de réduction ρ_2 est appelé un **résidu exact** du pas de réduction ρ_1 .
- Si les règles sont différentes, alors le pas de réduction ρ_2 est appelé un **faux résidu** du pas de réduction ρ_1 .

1. Ceci fonctionne d'ailleurs aussi en anglais, où le terme *redex*, contraction de *reducible expression*, indique une réduction possible sans postuler son unicité.

Exemple 7.2.

Considérons des symboles f, g, h, a , et b d'arités respectives 2, 1, 2, 0, et 0, ainsi que les règles de réécriture

$$\begin{aligned} (a) & : & a & \rightarrow b \\ (f_1) & : & f(Z_1, Z_2) & \rightarrow h(Z_1, Z_2) \\ (f_2) & : & f(b, Z) & \rightarrow g(Z) \end{aligned}$$

Le terme $t = f(a, b)$ est la source de deux pas de réduction :

- un pas $\rho_a : f(a, b) \rightarrow f(b, b)$ de position racine 1 pour la règle (a), et
- un pas $\rho_1 : f(a, b) \rightarrow h(a, b)$ de position racine ϵ pour la règle (f₁).

Considérons la cible $t' = f(b, b)$ du pas de réduction ρ_a . Elle est elle-même la source de deux pas de réduction de position racine ϵ (qui descend de ϵ) :

- un pas $\rho'_1 : f(b, b) \rightarrow h(b, b)$ pour la règle (f₁), qui est un résidu exact de ρ_1 , et
- un pas $\rho'_2 : f(b, b) \rightarrow g(b)$ pour la règle (f₂), qui est un faux résidu de ρ_1 .

Ainsi, la notion de « création d'un radical » n'est plus aussi pertinente qu'aux chapitres 5 et 6, puisqu'un descendant r' d'un radical r peut être associé à des pas de réduction très différents de ceux qui concernent r . Nous nous concentrerons à la place sur la création des pas de réduction.

Définition 7.3 (Création de pas de réduction). *Considérons un pas de réduction $\rho : t \rightarrow t'$. Un pas de réduction ρ' de source t' est **créé** par ρ s'il n'est le résidu exact après ρ d'aucun pas de réduction de source t .*

Exemple 7.4.

À l'exemple 7.2, le pas de réduction $\rho'_2 : f(b, b) \rightarrow g(b)$ est créé par le pas de réduction $\rho_a : f(a, b) \rightarrow f(b, b)$.

7.1.2 Création et modification de radicaux

Afin de préparer l'étiquetage causal des CRS, nous analysons dans cette section les manières dont un pas de réduction peut en créer un autre. Résumons d'abord les principaux éléments des raisonnements suivis dans les chapitres 5 et 6 pour la création de radicaux :

- Un pré-radical en λ -calcul ou dans le *Pure Pattern Calculus* est la conjonction d'une fonction avec la branche gauche d'une application. Nous devons donc enregistrer les événements qui ont mené à ce contact entre deux éléments. Sont associés les cas de création (1) et (2) qui créent un contact respectivement à la racine du réduit et à un point de substitution du réduit.
- Un pré-radical dans un système faible ne peut être un radical que si sa racine n'est pas gelée. Nous devons donc enregistrer les événements qui ont participé au dégel de la racine d'un radical. Est associé le cas de création (2⁺) qui dégèle une partie de l'épine de l'abstraction principale du radical. Ce cas recouvre intégralement le cas (2).

- Dans le *Pure Pattern Calculus*, un pré-radical n'est de plus un radical que si ses motif et argument ne sont de la bonne forme. Sont associés les cas de création (3) et (4) qui transforment « de l'intérieur » respectivement le motif et l'argument d'une opération de filtrage.

Chacun de ces trois éléments est déduit d'un aspect particulier de ce qu'est un radical. Essayons de raisonner de même dans les CRS.

Un radical dans un CRS est un sous-terme de la forme l^σ où l est le membre gauche d'une règle de réduction $l \rightarrow r$ et σ est une valuation sûre pour $l \rightarrow r$ (voir définitions 0.68 et 0.73 pour la sûreté). Cette définition suggère deux manières de contribuer à la création d'un radical : d'une part en contribuant à l'apparition du motif l (exemple 7.5) et d'autre part en contribuant à l'existence d'une valuation sûre σ (exemple 7.6).

Exemple 7.5.

Dans le λ -calcul vu comme un CRS, le membre gauche de la règle de β -réduction est le motif $@(\langle x \rangle Z(x), Z')$. Les cas de création (1) et (2) concernent tous deux l'apparition dans un terme du préfixe $@(\langle x \rangle \square, \square)$ de ce motif.

Exemple 7.6.

Considérons le CRS défini par les deux règles

$$\begin{aligned} g(Z_0) &\rightarrow a \\ \langle x \rangle f(Z_0, Z_1(x)) &\rightarrow Z_1(a) \end{aligned}$$

et regardons le terme

$$t = \langle x \rangle f(g(x), x)$$

La valuation

$$\sigma = \{Z_0 \mapsto \underline{\lambda} \emptyset . g(x), Z_1 \mapsto \underline{\lambda} x . x\}$$

n'est pas sûre pour la règle $\langle x \rangle f(Z_0, Z_1(x)) \rightarrow Z_1(a)$ et n'est donc pas acceptable. Le terme t n'est pas un radical. En revanche, t peut se réduire par la règle $g(Z_0) \rightarrow a$ en

$$t' = \langle x \rangle f(a, x)$$

Dans t' , le préfixe $\langle x \rangle f(\square, \square)$ correspondant aux positions propres du motif $\langle x \rangle f(Z_0, Z_1(x))$ n'a pas changé. En revanche la valuation

$$\sigma' = \{Z_0 \mapsto \underline{\lambda} \emptyset . a, Z_1 \mapsto \underline{\lambda} x . x\}$$

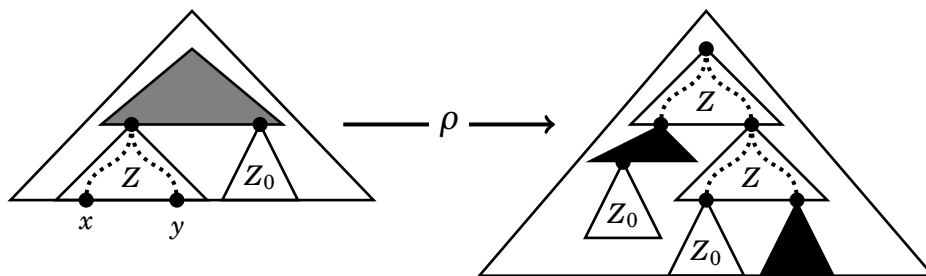
est maintenant sûre pour la règle $\langle x \rangle f(Z_0, Z_1(x)) \rightarrow Z_1(a)$, et le terme t' est un radical.

Cet exemple 7.6 montre comment, étant donné une règle $\kappa : l \rightarrow r$ et un terme t dans lequel le motif l apparaît mais tel qu'il n'existe pas de valuation σ sûre pour κ vérifiant $t = l^\sigma$, nous pouvons sans toucher les positions de t correspondant au motif l , obtenir un terme t' pour lequel il existe une valuation σ' sûre pour κ et telle que $t' = l^{\sigma'}$. Le passage d'un état à l'autre se fait par effacement d'une occurrence d'une variable liée. Comme nous considérons dans

ce chapitre encore des *CRS* faibles, ce cas ne sera finalement pas possible, et nous allons concentrer la suite de l'analyse sur la formation des motifs et sur le dégel des pré-radicaux gelés (le lemme 7.7 donnera *a posteriori* une justification formelle à ce resserrement du discours).

La question est donc : quelles actions opérées par un pas de réduction peuvent contribuer à former un motif? Dans le cadre du λ -calcul comme du *Pure Pattern Calculus*, l'action d'un pas de réduction se résumait à une réorganisation (avec duplication ou effacement éventuel) des symboles du radical touchant deux points : d'une part la racine du réduit où le corps de la fonction entrait en contact avec le contexte, et d'autre part les points de substitution où une position de l'argument entrait en contact avec une position du corps de la fonction.

Dans les *CRS* ces réorganisations de symboles sont toujours présentes, et côtoient en outre des créations de symboles. Ainsi, chaque position propre du membre droit d'une règle apporte au réduit de la structure neuve (représentée par les zones noires dans le schéma suivant) qui peut participer à la formation d'un motif. La « réorganisation de symboles » est elle apportée par les méta-variables du membre droit, et peut créer de nouveaux contacts comme le faisait la β -réduction. Quand une occurrence de méta-variable est remplacée par un substitut, deux interfaces de cette méta-variable sont importantes (représentée par des points noirs dans le schéma suivant). D'une part la racine du substitut entre en contact avec le contexte de la méta-variable, ou le contexte du réduit si la méta-variable est à la racine du membre droit de la règle considérée. D'autre part, aux points de substitution des arguments de la méta-variable dans le substitut les racines de ces arguments entrent en contact avec des positions du substitut, dont nous pouvons dire qu'elles sont des positions dynamiques (représentées par les chemins en pointillés ci-dessous).



Cette analyse nous pousserait donc à considérer comme positions critiques les positions propres du membre droit d'une règle (ce qui est un nouveau cas) ainsi que les racines des substituts et les racines de leurs arguments (qui correspondent respectivement aux cas (1) et (2) dans le λ -calcul).

À cette analyse de la création de pré-radicaux il faut ajouter, comme aux chapitres 5 et 6 précédents, les cas de création liés au dégel de certaines positions. Ces dégels peuvent toucher d'une part les positions propres du réduit, et d'autre part ses positions dynamiques. Les premières ont déjà été comptées, mais les deuxièmes s'ajoutent à la liste. De plus, comme dans le cas de création (2^+) aux chapitres 5 et 6, les positions dynamiques recouvrent l'un des autres effets, à

savoir la mise en contact entre les arguments d'une méta-variables et certaines positions dynamiques du substitut associé.

Les nouveautés apportées par un réduit concernent donc : ses positions propres, ses positions dynamiques, et les positions résultantes des positions des [racines des] méta-variables du membre droit de la règle utilisée. Nous pouvons cependant simplifier cet énoncé en considérant une dernière redondance. Considérons les cas possibles pour la position d'une méta-variable Z dans un membre droit de règle r .

- Z peut être immédiatement sous une position propre, $f(\dots, Z(\dots), \dots)$.
- Z peut être immédiatement sous une autre méta-variable, $Z'(\dots, Z(\dots), \dots)$
- Z peut être à la racine.

Dans les deux premiers cas, le contact entre la racine du substitut remplaçant Z et son contexte concerne un contexte qui est respectivement une position propre et une position dynamique. Il s'agit donc dans ces deux cas d'un contact avec une position qui est déjà par ailleurs considérée comme « critique ». Ne reste donc à considérer à part que le cas où la méta-variable est à la racine, dans lequel la racine du substitut entre en contact avec le contexte du réduit. Nous aboutissons à la classification formelle suivante.

Lemme 7.7 (Création d'un pas de réduction). *Soit $\rho : t = c[l^\sigma] \xrightarrow{p} c[r^\sigma] = t'$ un pas de réduction de position racine p . Soit ρ_c un pas de réduction de source t' qui n'est pas un résidu exact d'un pas de réduction de source t . Notons $l_c^{\sigma_c}$ le radical de ρ_c et p_c sa position. Alors l'ensemble des positions propres $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ du radical de ρ_c intersecte au moins l'un des ensembles suivants :*

- Le singleton $\{p\}$ contenant la racine de r^σ .
- Les positions propres $p \cdot \mathcal{P}_p(r, \sigma)$ de r^σ .
- Les positions dynamiques $p \cdot \mathcal{P}_d(r, \sigma)$ de r^σ .

Démonstration. Supposons que les positions propres $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ du radical créé n'intersectent ni les positions propres $p \cdot \mathcal{P}_p(r, \sigma)$ ni les positions dynamiques $p \cdot \mathcal{P}_d(r, \sigma)$ du réduit. Nous allons conclure en démontrant qu'alors la position racine p appartient aux positions propres $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ du radical créé.

Par notre hypothèse $p_c \cdot \mathcal{P}_p(l_c, \sigma_c) \subseteq \text{pos}(c) \cup p \cdot \mathcal{P}_s(r, \sigma)$. Si $p_c \cdot \mathcal{P}_p(l_c, \sigma_c) \subseteq \text{pos}(c)$ ou $p_c \cdot \mathcal{P}_p(l_c, \sigma_c) \subseteq \mathcal{P}_s(c, \sigma)$, alors l'ancêtre du radical $l_c^{\sigma_c}$ de ρ_c est déjà un radical pour la même règle, et ρ_c est un résidu exact du pas de réduction correspondant. Donc $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ intersecte à la fois $\text{pos}(c)$ et $p \cdot \mathcal{P}_s(r, \sigma)$. Or l_c est un motif (définition 0.61). Ainsi aucun symbole de l_c n'apparaît sous une méta-variable, et l'ensemble $\mathcal{P}_p(l_c, \sigma_c)$ est un segment initial (définition 0.25) : pour toute position $q_1 < q_2$, si $q_2 \in \mathcal{P}_p(l_c, \sigma_c)$ alors $q_1 \in \mathcal{P}_p(l_c, \sigma_c)$. Comme $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ intersecte $\text{pos}(c)$ nous avons $p_c < p$. Notons $p_c \cdot q_1 = p$. Comme $p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$ intersecte $p \cdot \mathcal{P}_s(r, \sigma)$ il existe une position $q_2 \in \mathcal{P}_s(r, \sigma)$ telle que $p \cdot q_2 \in p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$. Alors $p_c \cdot q_1 \cdot q_2 \in p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$, donc $q_1 \cdot q_2 \in \mathcal{P}_p(l_c, \sigma_c)$. Comme $\mathcal{P}_p(l_c, \sigma_c)$ est un segment initial nous avons donc $q_1 \in \mathcal{P}_p(l_c, \sigma_c)$, c'est-à-dire $p = p_c \cdot q_1 \in p_c \cdot \mathcal{P}_p(l_c, \sigma_c)$. \square

Le lemme 7.7 sur la création de pas de réduction isole deux parties importantes de la source et de la cible d'un pas de réduction : les positions propres du

radical, et la racine, les positions propres et les positions dynamiques du réduit. Ce lemme exprime au moyen d'une intersection les deux faits suivants :

- Toute cause directe de la création d'un pas de réduction touche les positions propres de son radical.
- Toute conséquence directe d'un pas de réduction touche à la zone définie par la racine, les positions propres, et les positions dynamiques de son réduit.

Cette analyse suggère un étiquetage fondé sur deux principes :

- Le nom d'une réduction est obtenu en collectant les étiquettes des positions propres de son radical.
- Le nom d'une réduction est diffusé dans toute la zone englobant la racine, les positions propres, et les positions dynamiques de son réduit.

7.2 Étiquetage

Cette section définit un étiquetage causal pour les *CRS* faibles, en mélangeant des aspects des étiquetages introduits au chapitre 4 pour les *CRS* et aux chapitres 5 et 6 pour le λ -calcul et le *Pure Pattern Calculus*. Les principes de base de cet étiquetage sont les suivants :

- Comme au chapitre 4 les étiquettes seront portées directement par les symboles de la signature et n'ajouteront pas de positions aux termes étiquetés. De plus l'ensemble du système étiqueté sera toujours un *CRS*, notamment grâce à l'utilisation du raffinement canonique.
- Comme aux chapitres 5 et 6 les pas de réduction porteront des noms construits avec toutes les étiquettes pertinentes de leur radical, et ces noms serviront de base pour déterminer les étiquettes du réduit.

Quant au choix des étiquettes et des modifications pertinentes, il suivra le résultat de l'analyse de la section précédente.

Commençons par délimiter les points sur lesquels l'étiquetage défini au chapitre 4 correspond ou ne correspond pas au comportement que nous voulons obtenir ici. Rappelons que les règles étiquetées (définition 4.42) ont la forme $l_\alpha \rightarrow [\tau(l_\alpha), i(r)]$, où :

- $\tau(\cdot)$ est une fonction récupérant l'étiquette racine d'un terme étiqueté (définition 1.13),
- $i(\cdot)$ est une fonction d'étiquetage initial (définition 4.39),
- l_α est un étiquetage d'un méta-terme l , et
- la paire $l \rightarrow r$ est une règle non-étiquetée appartenant au raffinement canonique d'une règle $l_0 \rightarrow r_0$ du système d'origine (Notons $l = l_0^{\sigma_0}$ et $r = r_0^{\sigma_0}$).

Observons d'abord que cet étiquetage du chapitre 4 n'utilise pas le nom Ω d'un pas de réduction ρ comme base des nouvelles étiquettes mais uniquement l'étiquette racine $\tau(l_\alpha)$ du radical l_α^σ de ρ . Il faudra donc ajouter un mécanisme de collecte pour former le nom Ω d'un pas de réduction ρ à partir des étiquettes de son radical. Ensuite, cet étiquetage agit sur les étiquettes des positions propres

du membre droit r d'une règle raffinée. Les positions propres $\mathcal{P}_p(r, \sigma)$ de ce membre droit r ont deux origines :

- Les positions propres $\mathcal{P}_p(r_0, \sigma \circ \sigma_0)$ du membre droit r_0 de la règle d'origine.
 - Les positions dynamiques $\mathcal{P}_d(r_0, \sigma \circ \sigma_0)$ du réduit selon la règle d'origine.
- Ainsi, par rapport à la règle $l_0 \rightarrow r_0$ du système d'origine, un pas de réduction étiqueté selon le chapitre 4 influe sur les étiquettes des positions propres et des positions dynamiques du réduit. Par conséquent nous couvrons deux des trois cas de création énoncés par le lemme 7.7 de création de pas de réduction.

En revanche, rien n'est explicitement prévu pour l'instant pour marquer la racine du réduit. Observons qu'un marquage spécifique de la racine est parfois utile.

Exemple 7.8.

Considérons la règle de réduction

$$@(\langle x \rangle Z(x), Z') \rightarrow Z(Z')$$

qui définit la β -réduction dans le cadre des CRS. La racine du membre droit $Z(Z')$ n'en étant pas une position propre, il n'est pas acquis que la racine du réduit d'un pas de réduction porte une étiquette modifiée. Les trois pas de réduction suivants montrent trois comportements possibles.

$$\begin{aligned} @^\omega(\langle x \rangle^\alpha @^\beta(x, x), a) &\rightarrow @^{[\omega, \beta]}(a, a) \\ @^\omega(\langle x \rangle^\alpha @^\beta(y, y), a) &\rightarrow @^\beta(y, y) \\ @^\omega(\langle x \rangle^\alpha x, a) &\rightarrow a \end{aligned}$$

Dans le premier cas la racine du substitut de Z en est une position dynamique, et son étiquette est donc modifiée. Dans le deuxième cas la racine est une position statique, et son étiquette n'est donc pas modifiée. Dans le troisième cas enfin le substitut de Z est dégénéré et ne porte aucune étiquette. Il s'agit d'un cas d'écrasement.

Comme l'a montré l'exemple 7.8 il existe des cas où aucune nouvelle étiquette ne marque la racine du réduit d'un pas de réduction. Dans ces conditions il est possible de perdre certaines informations de contribution, et en particulier de faire mentir le lemme 5.22 de contribution directe.

Exemple 7.9.

Considérons le terme étiqueté

$$@^\alpha(@^\beta(\langle y \rangle^\gamma y, \langle x \rangle^\delta t), u)$$

Le nœud $@^\beta$ définit un radical pour la β -réduction, par la contraction duquel nous obtenons le terme étiqueté

$$@^\alpha(\langle x \rangle^\delta t, u)$$

dans lequel nous venons de créer un nouveau radical pour la β -réduction selon le deuxième cas de Lévy (couvert par le cas de création (1) ici).

En revanche, comme le corps de la fonction du radical contracté ne contenait aucune étiquette, la cible $@^\alpha(\langle x \rangle^\delta t, u)$ ne porte aucune trace du pas de réduction. Ainsi le radical créé n'a aucune chance de porter la marque de son créateur.

Ces cas d'écrasement sont possibles dans le λ -calcul comme dans le *Pure Pattern Calculus*. Les étiquetages des chapitres 5 et 6 gèrent ces écrasements par l'étiquette $[\Omega]$ systématiquement ajoutée à la racine de tout radical étiqueté réduit. Dans les *CRS*, nous allons faire en sorte que la racine du réduit de tout pas de réduction en soit une position propre. Nous utiliserons pour cela la notion d'expansion d'un *CRS* présentée ci-dessous, qui est une transformation de *CRS*.

7.2.1 Expansion

Cette section définit l'*expansion* d'un *CRS*, une transformation de système inspirée de la transformation homonyme des systèmes du premier ordre proposée par van Oostrom [Ter03]. Cette transformation permet de ne pas perdre d'informations causales dans les systèmes avec *écrasement*, en particulier en forçant la racine de tout réduit à en être une position propre, et donc à porter une étiquette témoignant du pas de réduction qui vient d'avoir lieu.

Fixons un symbole distingué ι d'arité 1. Similairement aux étiquettes telles qu'employées aux chapitres 5 et 6, ce symbole sera utilisé uniquement en tant que support pour une information (qui sera une étiquette), et bien qu'il ajoute des positions aux termes il ne devra en aucun cas interférer avec les réductions du système. En fonction de l'étiquette qu'elle portera, une occurrence du symbole ι pourra en revanche influencer sur les noms des radicaux, et donc sur l'évolution des étiquettes en général.

Une expansion d'un terme t est obtenue en intercalant entre les positions de t un nombre arbitraire d'occurrences du symbole unaire ι . Les règles doivent être expansées pour permettre dans les termes expansés les mêmes réductions que dans les termes purs. Pour cela une règle $l \rightarrow r$ génère un ensemble de règles expansées dont les membres gauches couvrent toutes les expansions utiles de l . Le membre droit r aura en revanche une unique expansion. Notons la similitude avec l'ensemble des étiquetages d'une règle (définition 4.42), où tous les étiquetages du membre gauche sont considérés, contre uniquement l'étiquetage initial pour le membre droit.

Définition 7.10 (Compression et expansion). *Fixons un **symbole factice** distingué ι , qui est un symbole d'arité 1. Soit \mathcal{S} une signature de *CRS*. La **compression complète** $\chi(t_i)$ d'un méta-terme t_i sur $\mathcal{S} \sqcup \{\iota\}$ est le méta-terme sur \mathcal{S} défini par les équations suivantes :*

$$\begin{aligned} \chi(\iota(t)) &= \chi(t) \\ \chi(x) &= x \\ \chi([x]t) &= [x]\chi(t) \\ \chi(f(t_1, \dots, t_n)) &= f(\chi(t_1), \dots, \chi(t_n)) \\ \chi(Z(t_1, \dots, t_n)) &= Z(\chi(t_1), \dots, \chi(t_n)) \end{aligned}$$

Un méta-terme t_i sur $\mathcal{S} \sqcup \{i\}$ est une **expansion** d'un méta-terme t sur \mathcal{S} si $\chi(t_i) = t$. Le méta-terme t_i est une **expansion propre** du méta-terme t quand de plus, pour toute position $p \in \text{pos}(t_i)$ telle que $t_i(p) = i$ nous avons :

- $t_i(p \cdot 1)$ n'est pas une méta-variable, et
- si $p = q \cdot i$ alors $t_i(q)$ n'est pas une méta-variable.

En d'autres termes, une expansion propre est une expansion dont les symboles i ne sont pas adjacents à des méta-variables.

Exemple 7.11.

Le méta-terme $\iota @(\iota(\iota(\langle x \rangle Z(x))), Z')$ est une expansion propre du méta-terme $@(\langle x \rangle Z(x), Z')$. Le méta-terme $@(\langle x \rangle Z(\iota(x)), \iota(Z'))$ est une expansion de $@(\langle x \rangle Z(x), Z')$ mais n'est pas propre.

Définition 7.12 (Expansion d'une règle). Soit $\kappa : l \rightarrow r$ une règle d'un CRS $\Sigma = (\mathcal{S}, \mathcal{R})$. Notons $l = f(l_1, \dots, l_n)$. L'ensemble des **expansions** de la règle $\kappa : l \rightarrow r$ est l'ensemble de règles

$$E(\kappa) = \{f(l_1^e, \dots, l_n^e) \rightarrow \iota(r) \mid \text{pour tout } i, l_i^e \text{ est une expansion propre de } l_i\}$$

Définition 7.13 (Expansion d'un système). L'**expansion** d'un CRS $\Sigma = (\mathcal{S}, \mathcal{R})$ est le CRS $E(\Sigma)$ dont la signature est $\mathcal{S} \cup \{i\}$ et l'ensemble de règles est l'union des expansions des règles de \mathcal{R} .

Nous pourrions vérifier à la section 7.3 que l'expansion $E(\Sigma)$ d'un système Σ a le même comportement que Σ , dans le sens que les deux systèmes sont bisimilaires (lemme 7.34) et que la transformation préserve le partage pleinement paresseux (théorèmes 7.29 et 7.36), l'éventuelle orthogonalité du système (lemme 7.39), et la notion de descendance (lemme 7.38).

7.2.2 Étiquetage

Nous pouvons maintenant définir une variante de l'étiquetage proposé au chapitre 4. Une fois établi le mécanisme d'attribution de noms aux pas de réduction, le développement de cette section sera identique à la section 4.3. Comme au chapitre 6, le principal enjeu est donc de déterminer des noms pour les pas de réduction. Cependant, le formalisme des CRS est beaucoup plus « explicite » que le *Pure Pattern Calculus*, dans le sens que toutes les contraintes d'application d'une règle de réécriture $\kappa : l \rightarrow r$ sont visibles dans cette règle, et que nous n'avons pas comme avec le filtrage composé de conditions additionnelles définies au niveau méta. Les noms seront donc simplement obtenus en collectant les étiquettes du membre gauche de la règle appliquée.

Commençons par définir un ensemble d'étiquettes similaire à celui utilisé aux chapitres 5 et 6.

Définition 7.14 (Noms et étiquettes). L'ensemble d'étiquettes \mathcal{L} et l'ensemble de noms de radicaux \mathcal{N} sont définis par la grammaire suivante :

$$\begin{aligned} \alpha & ::= p \mid [\Omega, p] && p \text{ position} \\ \Omega & ::= \alpha_1 \dots \alpha_n && n \geq 1 \end{aligned}$$

Nous pouvons maintenant recopier les définitions de la section 4.3 en utilisant le nouvel ensemble d'étiquettes.

Définition 7.15 (Signature étiquetée). *Soit \mathcal{S} une signature CRS. La **signature étiquetée** $\mathcal{S}^{\mathcal{L}}$ est définie par*

$$\mathcal{S}^{\mathcal{L}} = \bigcup_n \mathcal{S}_n^{\mathcal{L}}$$

avec pour toute arité n

$$\mathcal{S}_n^{\mathcal{L}} = \{f^\alpha \mid f \in \mathcal{S}_n, \alpha \in \mathcal{L}\}$$

Nous notons $(.)^\bullet : \mathcal{S}^{\mathcal{L}} \rightarrow \mathcal{S}$ la fonction d'oubli des étiquettes telle que pour tous $f \in \mathcal{S}$ et $\alpha \in \mathcal{L}$

$$(f^\alpha)^\bullet = f$$

Définition 7.16 (Étiquetage d'un méta-terme). *Soit \mathcal{S} une signature CRS. La fonction de désétiquetage $(.)^\bullet$ induit un morphisme des méta-termes sur $\mathcal{S}^{\mathcal{L}}$ vers les méta-termes sur \mathcal{S} . Considérons un méta-terme t^l sur $\mathcal{S}^{\mathcal{L}}$ et un méta-terme t sur \mathcal{S} . Nous disons que t^l est un **étiquetage** de t si $(t^l)^\bullet = t$.*

Définition 7.17 (Étiquetage initial). *Soit \mathcal{S} une signature CRS. Pour toute position p la fonction $i_p(.)$ transforme un terme sur \mathcal{S} en un terme sur $\mathcal{S}^{\mathcal{L}}$ par les règles suivantes :*

$$\begin{aligned} i_p(x) &= x \\ i_p([x]t) &= [x]i_p(t) \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p.1}(t_1), \dots, i_{p.n}(t_n)) \\ i_p(Z(t_1, \dots, t_n)) &= Z(i_{p.1}(t_1), \dots, i_{p.n}(t_n)) \end{aligned}$$

Soit t un méta-terme sur une signature \mathcal{S} . L'**étiquetage initial** de t est le méta-terme $i_\epsilon(t)$, que nous notons encore $i(t)$.

Définition 7.18 (Réétiquetage uniforme). *Pour tout terme t sur une signature $\mathcal{S}^{\mathcal{L}}$ et séquence d'étiquettes Ω , le réétiquetage uniforme $[\Omega, t]$ est défini ainsi :*

$$\begin{aligned} [\Omega, x] &= x \\ [\Omega, [x]t] &= [x][\Omega, t] \\ [\Omega, f^p(t_1, \dots, t_n)] &= f^{[\Omega, p]}([\Omega, t_1], \dots, [\Omega, t_n]) \\ [\Omega, Z(t_1, \dots, t_n)] &= Z([\Omega, t_1], \dots, [\Omega, t_n]) \end{aligned}$$

Notons qu'à ce chapitre le réétiquetage uniforme ne sera appliqué qu'à des termes munis d'un étiquetage initial, nous ne perdons donc rien à ne considérer que des positions dans le troisième cas de la définition.

Exemple 7.19.

Considérons le méta-terme

$$t = i(@ (i(i(\langle x \rangle Z(x))), Z'))$$

Son étiquetage initial est

$$i(t) = \iota^\epsilon(@^1(\iota^{11}(\iota^{111}(\langle x \rangle^{1111}Z(x))), Z'))$$

et le renommage uniforme par $\alpha\beta$ de cet étiquetage initial est

$$[\alpha\beta, i(t)] = \iota^{[\alpha\beta, \epsilon]}(@^{[\alpha\beta, 1]}(\iota^{[\alpha\beta, 11]}(\iota^{[\alpha\beta, 111]}(\langle x \rangle^{[\alpha\beta, 1111]}Z(x))), Z'))$$

Au chapitre 4, une règle de réduction étiquetée a la forme $l_\alpha \rightarrow [\tau(l_\alpha), i(r)]$, dans laquelle les étiquettes du membre droit $[\tau(l_\alpha), i(r)]$ sont paramétrées par l'étiquette racine $\tau(l_\alpha)$ du membre gauche l_α . Ici, nous utiliserons à la place de cette étiquette racine une séquence d'étiquettes prélevées dans l_α , qui contient toutes les étiquettes des symboles qui ne sont pas en dessous d'une méta-variable.

Définition 7.20. Pour tout méta-terme étiqueté t , nous notons $\|t\|$ la séquence d'étiquettes définie par :

$$\begin{aligned} \|x\| &= \epsilon \\ \|[x]t\| &= \alpha\|t\| \\ \|f^\alpha(t_1, \dots, t_n)\| &= \alpha\|t_1\| \dots \|t_n\| \\ \|Z(t_1, \dots, t_n)\| &= \epsilon \end{aligned}$$

Le nom d'un pas de réduction pour une règle étiquetée de membre gauche l_α sera la séquence fournie par $\|l_\alpha\|$.

Définition 7.21 (Règle étiquetée). Soit $\kappa : l \rightarrow r$ une règle CRS. L'ensemble $\kappa^\mathcal{L}$ des **étiquetages de la règle** $\kappa : l \rightarrow r$ est l'ensemble de règles étiquetées

$$\kappa^\mathcal{L} = \{ l_\alpha \rightarrow [\|l_\alpha\|, i(r)] \mid l_\alpha^\bullet = l \}$$

Si \mathcal{R} est un ensemble de règles CRS, alors nous notons $\mathcal{R}^\mathcal{L}$ l'ensemble des étiquetages des règles de \mathcal{R} .

Enfin, nous pouvons définir l'étiquetage causal d'un CRS faible Σ en prenant l'expansion $E(\Sigma)$ de Σ , puis le raffinement canonique $R(E(\Sigma))$ de cette expansion (ainsi qu'il a été fait au chapitre 4 pour l'étiquetage pleinement paresseux), puis enfin en appliquant la discipline d'étiquetage décrite dans cette section.

Définition 7.22 (Étiquetage causal). Soit un CRS faible $\Sigma = (\mathcal{S}, \mathcal{R})$. Son **étiquetage causal** est le CRS $R(E(\Sigma))^\mathcal{L}$ dont la signature est $(\mathcal{S} \sqcup \{t\})^\mathcal{L}$ et l'ensemble de règles est $R(E(\Sigma))^\mathcal{L}$.

Remarque 7.23. Nous avons opté ici pour une solution simpliste, qui ne transcrit pas directement le lemme 7.7 de création de pas de réduction. En effet, comme au chapitre 4, les étiquetages que nous considérerons pour une règle $\kappa_0 : l_0 \rightarrow r_0$ du système d'origine seront les étiquetages des raffinements canoniques de κ_0 .

Ainsi les étiquettes collectées ne seront pas celles des positions propres du membre gauche de la règle d'origine κ_0 , mais les positions propres du membre gauche d'une règle raffinée.

Cette variante est cependant acceptable d'un point de vue des propriétés de causalité. Ceci pourrait être informellement justifié a priori à la lumière des discussions menées à la section 5.6 sur l'anticipation et le retardement des propagations d'informations causales. Plus prosaïquement, nous vérifierons la pertinence de cette variante a posteriori, aux sections 7.4 et 7.5, en dérivant les propriétés causales du système obtenu.

Enfin, comme aux chapitres 5 et 6, nous définissons sur les séquences d'étiquettes une relation de *contribution directe* et une relation de *contribution*, qui sont censées refléter les dépendances entre pas de réduction.

Définition 7.24 (Contribution). Une séquence d'étiquettes Ω **contribue directement** à une séquence d'étiquettes Γ , noté $\Omega \hookrightarrow^d \Gamma$, s'il existe deux séquences d'étiquettes éventuellement vides Γ_1 et Γ_2 telles que

$$\Gamma = \Gamma_1[\Omega, p]\Gamma_2$$

La relation de **contribution** est la clôture transitive de \hookrightarrow^d .

7.3 Équivalence avec le partage pleinement paresseux

Nous allons maintenant vérifier que l'étiquetage causal des CRS faibles définit un SLS dont la réduction étiquetée parallèle est bisimilaire à la réduction avec partage pleinement paresseux définie au chapitre 4. Cette vérification est effectuée en trois temps : à la section 7.3.1 nous vérifions que l'étiquetage causal définit un SLS, puis à la section 7.3.2 nous montrons que l'étiquetage causal d'un CRS Σ et l'étiquetage pleinement paresseux de l'expansion $E(\Sigma)$ définissent des réductions parallèles étiquetées bisimilaires, et enfin à la section 7.3.3 nous montrons que l'étiquetage pleinement paresseux de l'expansion $E(\Sigma)$ et du système Σ brut définissent des réductions parallèles étiquetées bisimilaires.

7.3.1 Propriété de partage

Nous pouvons maintenant recopier la formalisation de la représentation de notre CRS étiqueté en un SLS effectuée au chapitre 4, en considérant l'expansion du système d'origine avant d'en prendre le raffinement canonique et en remplaçant aux endroits adéquats l'étiquette $\tau(l_\alpha)$ par la séquence $\|l_\alpha\|$. Ce dernier point implique également de prendre un ordre sur les étiquettes adapté des chapitres 5 et 6.

Définition 7.25 (Ordre sur les étiquettes). L'ensemble \mathcal{L} d'étiquettes de la définition 4.35 est muni de l'ordre partiel \leq généré par :

$$\omega_1 \leq [\omega_1 \dots \omega_n, p]$$

Instanciation 7.26. Soit un CRS faible $(\mathcal{S}, \mathcal{R})$. Considérons l'étiquetage $((\mathcal{S} \sqcup \{t\})^{\mathcal{L}}, R(E(\mathcal{R}))^{\mathcal{L}})$ de son raffinement canonique et définissons :

- \mathcal{T} l'ensemble des termes syntaxiques sur $\mathcal{X} \cup \{x \mid x \in \mathcal{X}\} \cup (\mathcal{S} \sqcup \{t\})^{\mathcal{L}}$.
- \mathcal{R} l'ensemble des pas de réduction faible par les règles de $R(E(\mathcal{R}))^{\mathcal{L}}$.

Soit $\rho \in \mathcal{R}$. Notons $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et définissons :

- $\text{dex}(\rho) = l^\sigma$
- $\text{duct}(\rho) = r^\sigma$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \mathcal{P}_p(r, \sigma)$

Notons $\Sigma = (\mathcal{S}[\mathcal{L}] \cup \mathcal{X}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$.

Lemme 7.27 (ATRS marqué). *Le système Σ est un ATRS marqué.*

Démonstration.

Source & Cible Tout terme syntaxique est dans \mathcal{T} .

Résidus Soient deux pas de réduction $\rho_1, \rho_2 \in \mathcal{R}$ tels que $\text{src}(\rho_1) = \text{src}(\rho_2)$ et les positions racines $\text{root}(\rho_1)$ et $\text{root}(\rho_2)$ sont disjointes. Notons $\rho_1 : c_1[l_1^{\sigma_1}] \rightarrow c_1[r_1^{\sigma_1}]$ et $\rho_2 : c_2[l_2^{\sigma_2}] \rightarrow c_2[r_2^{\sigma_2}]$. Comme $\text{root}(\rho_1)$ et $\text{root}(\rho_2)$ sont disjointes, il existe c un contexte binaire tel que $\text{src}(\rho_1) = \text{src}(\rho_2) = c[l_1^{\sigma_1}, l_2^{\sigma_2}]$ et $\text{tgt}(\rho_1) = c[r_1^{\sigma_1}, l_2^{\sigma_2}]$. Alors $\rho'_2 : \text{tgt}(\rho_1) \rightarrow c[r_1^{\sigma_1}, r_2^{\sigma_2}]$ est un pas de réduction de \mathcal{R} équivalent à ρ_2 .

Zone d'effet Soient $\rho \in \mathcal{R}$ et $q \in \text{pos}(\text{duct}(\rho)) \setminus \text{effz}(\rho)$. Par définition de effz , $q \in \mathcal{P}_s(r, \sigma)$. Conclusion par lemme 4.44. \square

Lemme 7.28 (SLS). *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration.

Étiquette racine Par définition de \mathcal{T} , tous les symboles sauf les variables sont étiquetées, et le symbole de tête d'un radical ne peut être une variable.

Progression Soit un pas de réduction $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$. Toute étiquette de la zone d'effet de ρ est une étiquette de r , c'est-à-dire une étiquette de la forme $[\|l\|, p]$, où par définition la séquence d'étiquettes $\|l\|$ commence par $\tau(l) = \tau(\rho)$. Donc $\tau(\rho) \leq [\|l\|, p]$.

Héritage Soient $\rho \in \mathcal{R}$ et $\alpha, \beta \in \mathcal{L}$ telles que β apparaît dans $\text{effz}(\rho)$, $\alpha < \beta$ et $\alpha \neq \tau(\rho)$. En particulier β est de la forme $[\omega_1 \dots \omega_n, p]$ avec $\tau(\rho) = \omega_1$, donc $\alpha \leq \tau(\rho)$ et même $\alpha < \tau(\rho)$. De plus $\tau(\rho)$ apparaît dans $\text{dex}(\rho)$.

Partage L'étiquette à la position p dans la zone d'effet d'un pas de réduction ρ de nom Ω est $[\Omega, p]$. Toutes les positions de la zone d'effet portent donc des étiquettes différentes. \square

Ainsi, comme dans les cadres du λ -calcul faible (chapitre 5) et du *Pure Pattern Calculus* faible (chapitre 6), l'étiquetage « causal » des CRS faibles définit un SLS, c'est-à-dire une réduction partagée. Nous allons maintenant établir l'équivalence entre cette réduction partagée et le partage pleinement paresseux défini au chapitre 4.

Cette équivalence, qui avait seulement été évoquée pour le λ -calcul faible (à la section 5.5) est ici décomposée en deux étapes :

- La section 7.3.2 montre d’abord, dans un cadre avec expansion, que créer les étiquettes du réduit à l’aide d’un nom de radical complexe (comme au présent chapitre) ou à l’aide seulement de l’étiquette racine du radical (comme au chapitre 4) donne le même partage.
- La section 7.3.3 montre ensuite, dans un cadre où les étiquettes du réduit sont construites à l’aide de l’étiquette racine du radical (comme au chapitre 4), que l’expansion n’influe pas sur le partage.

L’avantage de considérer cette décomposition, et dans cet ordre, vient de l’interaction entre l’expansion et le nommage des pas de réduction qui existe dans les définitions de la section 7.2 : ajouter ou enlever des expansions dans un membre gauche de règle modifie le nom des pas de réduction associés. En revanche, comme nous n’expansons pas la racine des membres gauches de règles (définition 7.12), si les étiquettes du réduit ne sont construites qu’à partir de l’étiquette racine du radical alors l’expansion ne change rien aux étiquettes.

7.3.2 Pleine paresse avec expansion

Prenons un CRS Σ , et considérons le raffinement canonique de son expansion, qui est un CRS noté $R(E(\Sigma))$. Sur ce système, considérons l’étiquetage causal selon la section 7.2 et l’étiquetage pleinement paresseux selon la section 4.3. À l’instar de ce que nous avons vu à la section 4.4 quant à la relation entre les étiquetages que donnent les chapitres 2 et 4 du λ -calcul, nous pouvons résumer la relation entre ces deux CRS étiquetés en trois points :

- Ils sont deux étiquetages du même *ATRS* sous-jacent.
- Ils sont différents.
- Ils définissent des relations de réduction parallèle étiquetée bisimilaires.

La conjonction des premier et troisième points donne l’équivalence de ces deux notions de partage. Les propriétés communes qui permettent d’établir cette équivalence du partage sont encore les deux suivantes :

- Les pas de réduction des deux systèmes ont les mêmes zones d’effet.
- Les étiquettes de la zone d’effet de tout pas de réduction sont toutes différentes.

L’énoncé de cette équivalence utilise encore la notion de termes miroirs introduite à la section 1.5.

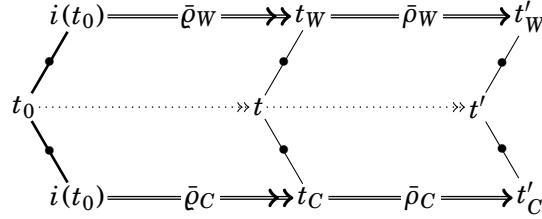
Théorème 7.29 (Partage équivalent). *Soient un CRS Σ et le CRS $R(E(\Sigma))$ obtenu par expansion et raffinement canonique de Σ .*

Soient t_W un terme étiqueté de $R(E(\Sigma))$ selon le chapitre 4 et t_C un terme étiqueté selon le présent chapitre tels que $t_W \approx t_C$. Alors t_W et t_C induisent la même relation d’équivalence par étiquettes (définition 3.25) sur les positions de $t = t_W^\bullet = t_C^\bullet$.

Démonstration. Par définition de $t_W \approx t_C$, il existe un terme non étiqueté t_0 et une séquence de réduction miroir parallèle $(\bar{\rho}_W, \bar{\rho}_C)$ de source $(i(t_0), i(t_0))$ et de cible (t_W, t_C) . Raisonnons par récurrence sur le nombre d’étapes de réduction parallèle étiquetée dans la séquence.

Le cas de base, avec une séquence de longueur 0, est immédiat car alors $t_W = i(t_0) = t_C$.

Supposons donc que la propriété est vérifiée pour les séquences de réduction miroir parallèle de longueur n , et considérons une séquence de réduction miroir parallèle $(\bar{\rho}_W \bar{\rho}_W, \bar{\rho}_C \bar{\rho}_C)$ de cible (t'_W, t'_C) , avec $(\bar{\rho}_W, \bar{\rho}_C)$ une séquence de réduction miroir parallèle de longueur n . Notons t_W et t_C les cibles respectives de $\bar{\rho}_W$ et $\bar{\rho}_C$.



Par hypothèse de récurrence les termes t_W et t_C induisent la même relation d'équivalence par étiquettes sur les positions de $t = t_W = t_C$. De plus, t_W et t_C vérifient tous deux les propriétés \mathbb{S} et \mathbb{I} par lemme 1.18 et théorème 1.19. Notons ω_W l'étiquette racine des radicaux réduits en parallèle par $\bar{\rho}_W$ et Ω_C le nom des radicaux réduits en parallèle par $\bar{\rho}_C$.

Soient p' et q' deux positions de t' . Notons p et q leurs ancêtres respectifs dans t . Supposons que $\tau_{p'}(t'_W) = \tau_{q'}(t'_W) = \alpha$. Nous allons montrer que $\tau_{p'}(t'_C) = \tau_{q'}(t'_C)$. Raisonnons par cas sur les positions p' et q' .

- Si ni p' ni q' n'est dans la zone d'effet d'un des pas de réduction parallèle, alors $\tau_p(t_W) = \tau_{p'}(t'_W) = \tau_{q'}(t'_W) = \tau_q(t_W)$, donc par hypothèse de réduction $\tau_p(t_C) = \tau_q(t_C)$, et finalement $\tau_{p'}(t'_C) = \tau_p(t_C) = \tau_q(t_C) = \tau_{q'}(t'_C)$.
- Si p' est dans la zone d'effet d'un des pas de réduction parallèle et pas q' , alors $\alpha = \tau_{p'}(t'_W)$ est de la forme $[\omega_W, p_e]$ (où p_e est la position à l'intérieure de la zone d'effet) et $\alpha = \tau_{q'}(t'_W) = \tau_q(t_W)$, donc $\omega_W \leq \tau_q(t_W)$, ce qui contredit $\mathbb{I}(t_W)$.
- Si p' et q' sont tous deux dans la zone d'effet d'un des pas de réduction parallèle, alors $\alpha = \tau_{p'}(t'_W)$ est de la forme $[\omega_W, p_e]$ et $\alpha = \tau_{q'}(t'_W)$ est de la forme $[\omega_W, q_e]$, avec en particulier $p_e = q_e$, et donc $\tau_{p'}(t'_C) = [\Omega_C, p_e] = [\Omega_C, q_e] = \tau_{q'}(t'_C)$.

Nous montrons similairement que $\tau_{p'}(t'_C) = \tau_{q'}(t'_C)$ implique $\tau_{p'}(t'_W) = \tau_{q'}(t'_W)$, ce qui conclut la preuve. \square

Ainsi, baser la génération des nouvelles étiquettes sur des noms complexes issus des radicaux ne change rien au partage. Nous pouvons d'ailleurs nous y attendre : si la propriété de partage \mathbb{S} est vérifiée, alors deux radicaux de même étiquette racine sont égaux et ont donc même nom.

7.3.3 De la transparence de l'expansion

Pour finaliser le lien entre l'étiquetage causal d'un CRS et son étiquetage pleinement paresseux, nous allons maintenant vérifier que l'expansion du sys-

tème n'affecte pas ses réductions parallèles étiquetées (théorème 7.36). D'une manière plus générale, nous allons dans cette section montrer que la compression (l'inverse de l'expansion, définition 7.10) préserve l'essentiel de ce qui nous intéresse du comportement des systèmes, à savoir les séquences de réduction et leurs longueurs (lemme 7.34 de bisimulation), la descendance (lemme 7.38), et l'éventuelle orthogonalité des systèmes (lemme 7.39). Ces propriétés mises ensemble nous permettront à terme d'assurer également la préservation des notions de résiduation, de nécessité, et finalement de partage optimal.

Ces propriétés sont essentiellement immédiates dans le cadre de l'expansion/compression, mais elles permettent de mettre en place la méthode qui sera utilisée aux chapitres 8 et 9 pour montrer les bonnes propriétés de la transformation de *lifting*.

Cette méthode est basée sur l'utilisation d'une définition alternative « à petits pas » d'une transformation qui nous est donnée par une fonction globale (ici la fonction $\chi(\cdot)$ de compression). Cette définition se fait dans le cadre d'un système de réécriture dans lequel peuvent être exprimés à la fois le système source (ici un CRS expansé), le système cible (ici le même CRS, mais compressé), et la transformation (ici la compression, vue comme une suite de pas de réécriture).

La représentation « à petit pas » permet de ramener la preuve de correction de la transformation globale à la preuve de correction d'un seul pas de transformation. Nous pourrions donc nous contenter d'analyses locales et simples. L'utilisation d'un cadre de réécriture pour représenter la transformation permet de plus d'utiliser des notions existantes comme la descendance pour établir un lien entre les positions des termes avant et après transformation.

Considérons un CRS Σ , son expansion $E(\Sigma)$, le raffinement canonique $R(E(\Sigma))$, et l'étiquetage pleinement paresseux $R(E(\Sigma))^{\mathcal{L}}$ selon le chapitre 4.

Définition 7.30 (Règle de compression). *Nous pouvons ajouter à $E(\Sigma)$ une règle c dite de **compression***

$$c : \iota(Z) \rightarrow_c Z$$

dont l'unique raffinement est elle-même, et donc les étiquetages sont donnés par le schéma de règle suivant pour toute étiquette $\alpha \in \mathcal{L}$:

$$c_\alpha : \iota^\alpha(Z) \rightarrow_c Z$$

*Un **pas de compression** est l'application d'une règle de compression dans un contexte arbitraire (nous ne considérons aucune restriction à la réduction faible ici).*

Du point de vue de l'interprétation graphique, la compression peut être comprise comme la suppression des indirections que représentent les nœuds factices $\iota^\alpha(\cdot)$. Vérifions maintenant que ces règles de réécriture décrivent sans ambiguïté une fonction (lemme 7.31 de convergence) et que cette fonction coïncide bien avec la fonction de compression de la définition 7.10 (lemme 7.32).

Lemme 7.31 (Convergence). *Les règles de compression définissent un système de réécriture convergent (c'est-à-dire confluent et fortement normalisant).*

Démonstration. Le système défini par les règles de compression est orthogonal et donc confluent ([KvOvR93]). De plus, il n'existe aucune séquence de compression infinie, puisque chaque pas fait décroître d'une unité le nombre d'occurrences dans le terme de symboles de la forme ι^α . \square

Le système formé par les règles de compression est donc convergent, et définit pour chaque terme une unique forme normale. Les formes normales étant caractérisées en ce qu'elles ne contiennent aucun symbole de la forme ι^α , elles appartiennent au système $R(\Sigma)^\mathcal{L}$. Les règles de compression définissent donc une fonction des termes de $R(E(\Sigma))^\mathcal{L}$ dans les termes de $R(\Sigma)^\mathcal{L}$, dont le lemme 7.32 de correction suivant vérifie qu'elle est bien la fonction de compression $\chi(\cdot)$ de la définition 7.10.

Lemme 7.32 (Correction des règles de compression). *Soit t un méta-terme de $R(E(\Sigma))^\mathcal{L}$. Alors $\chi(t)$ est la forme normale de t par les règles de compression.*

Démonstration. Par induction sur t nous pouvons montrer que $t \rightarrow_c \chi(t)$ par une séquence de pas de compression. Or $\chi(t)$ est une forme normale pour les règles de compression, d'où conclusion par lemme 7.31 de convergence. \square

Ainsi nous avons deux caractérisations équivalentes d'un lien entre les termes de $R(E(\Sigma))^\mathcal{L}$ et les termes de $R(\Sigma)^\mathcal{L}$, l'une basée sur une fonction de compression définie au niveau méta, l'autre basée sur une règle de réécriture ajoutée au système. Pour démontrer que cette compression définit une bisimulation entre les réductions parallèles étiquetées des deux systèmes nous allons utiliser la caractérisation basée sur la réécriture.

Nous allons bénéficier du fait que les règles expansées (définition 7.12) sont définies indépendamment du nombre d'occurrences du symbole ι . En particulier, les pas de compression ne sont pas seulement une transformation progressive des termes, mais aussi une transformation progressive des séquences de réduction.

Lemme 7.33 (Commutation).

$$\begin{array}{l}
 - \text{ Si } \begin{array}{ccc} t & \xrightarrow{c} & u \\ \downarrow & & \\ t' & & \end{array} \text{ alors il existe } u' \text{ tel que } \begin{array}{ccc} & & u \\ & & \downarrow \\ t' & \xrightarrow{c} & u' \end{array} . \\
 - \text{ Si } \begin{array}{ccc} t & \xrightarrow{c} & u \\ & & \downarrow \\ & & u' \end{array} \text{ alors il existe } t' \text{ tel que } \begin{array}{ccc} & & t \\ \downarrow & & \\ t' & \xrightarrow{c} & u' \end{array} .
 \end{array}$$

Démonstration. Par cas sur les position relatives des radicaux des pas de réduction $t \rightarrow u$ et $t \rightarrow t'$ (ou $u \rightarrow u'$). L'argument principal est que l'expansion d'une règle (définition 7.12) admet une répartition quelconque des symboles ι . La séquence $t' \rightarrow_c u'$ vient du fait qu'un pas de réduction peut dupliquer un symbole ι . \square

Ce lemme de bisimulation simple a pour corollaire :

Lemme 7.34 (Bisimulation). *La fonction $\chi(\cdot)$ définit une bisimulation entre les systèmes $R(E(\Sigma))^{\mathcal{L}}$ et $R(\Sigma)^{\mathcal{L}}$.*

Démonstration. Par lemme 7.32 de correction, pour tout terme t du système $R(E(\Sigma))^{\mathcal{L}}$ il existe une séquence de compression $t \rightarrow_c \chi(t)$. Par récurrence sur la longueur de cette séquence, le lemme 7.33 permet de conclure. \square

Nous pouvons de plus observer que ce résultat ne concerne que l'expansion, et est indépendant du fait que le système puisse être raffiné ou étiqueté.

Enfin, le résultat de bisimulation 7.34 peut être associé à une remarque de préservation des étiquettes pour en déduire que la bisimulation vaut également pour la réduction parallèle étiquetée.

Lemme 7.35 (Préservation du partage). *Soit $\bar{\rho} : t \Rightarrow_c t'$ une étape de compression parallèle étiquetée. Soient deux positions $p', q' \in \text{pos}(t')$. Alors $\tau_{p'}(t') = \tau_{q'}(t')$ si et seulement si les ancêtres respectifs p et q de p' et q' dans t vérifient $\tau_p(t) = \tau_q(t)$.*

Démonstration. Immédiat, la compression ne modifiant aucune étiquette. \square

Théorème 7.36 (Bisimulation pour la réduction partagée). *La fonction $\chi(\cdot)$ définit une bisimulation entre les réductions parallèles étiquetées des systèmes $R(E(\Sigma))^{\mathcal{L}}$ et $R(\Sigma)^{\mathcal{L}}$.*

Démonstration. Le lemme 7.35 permet, sous les hypothèses de la propriété de partage \mathbb{S} et de la propriété d'indépendance \mathbb{I} , d'exprimer le lemme 7.33 avec la réduction parallèle étiquetée et la compression parallèle étiquetée. De là nous déduisons la bisimulation. \square

La préservation de la descendance par compression peut être déduite des propriétés précédentes grâce à une astuce liée aux étiquettes. Ces dernières permettent en effet de suivre les relations de descendance.

Lemme 7.37 (Descendance et étiquettes). *Considérons un CRS Σ (qui peut contenir la règle de compression), son expansion $E(\Sigma)$, le raffinement canonique $R(E(\Sigma))$ et l'étiquetage pleinement paresseux $R(E(\Sigma))^{\mathcal{L}}$. Soit t un terme étiqueté dont toutes les étiquettes sont initiales et différentes, et $t \rightarrow t'$ une séquence de réduction étiquetée. Alors une position $p' \in \text{pos}(t')$ descend d'une position $p \in \text{pos}(t)$ si et seulement si $\tau_{p'}(t') = \tau_p(t)$.*

Démonstration. Par récurrence sur la longueur de la séquence, avec l'énoncé généralisé suivant :

Pour tout terme étiqueté t , toute séquence de réduction $\rho : t \rightarrow t'$, et toute étiquette α apparaissant dans t , notons $P_\alpha = \{p \in \text{pos}(t) \mid \tau_p(t) = \alpha\}$ et $P'_\alpha = \{p' \in \text{pos}(t') \mid \tau_{p'}(t') = \alpha\}$ les ensembles des occurrences de α dans t et t' . Alors $P'_\alpha = P_\alpha / \rho$. \square

La descendance étant liée aux étiquettes, nous pouvons nous servir des résultats de confluence sur la réduction étiquetée pour déduire la stabilité par compression de la notion de descendance.

Lemme 7.38 (Préservation de la descendance). *Considérons un CRS Σ , et $E(R(\Sigma))$ le raffinement canonique de son expansion. Soit un pas de compression $\rho_c : t \rightarrow_c u$ et un pas de réduction $\rho_t : t \rightarrow t'$, et notons $\rho_u : u \rightarrow u'$ et $\rho'_c : t' \rightarrow_c u'$ la clôture du diagramme selon le lemme 7.33 de bisimulation simple. Alors les séquences $\rho_c \rho_u$ et $\rho_t \rho'_c$ définissent la même relation de descendance entre les positions de t et de u' .*

Démonstration. Considérons un étiquetage t_l de t par des étiquettes toutes initiales et différentes. Par lemme 7.33 les versions étiquetées des séquences $\rho_c \rho_u$ et $\rho_t \rho'_c$ ont toujours une même cible u'_l . Alors par lemme 7.37 liant la descendance aux étiquettes, nous pouvons conclure. \square

Enfin, nous pouvons vérifier que l'expansion préserve l'orthogonalité des systèmes.

Lemme 7.39 (Expansion et orthogonalité). *Soit Σ un CRS. Alors le CRS $E(\Sigma)$ est orthogonal si et seulement si Σ l'est aussi.*

Démonstration. La preuve est similaire à celle effectuée par van Oostrom pour l'expansion de systèmes du premier ordre [Ter03, Chap. 8]. \square

La technique d'analyse utilisée dans cette section, qui caractérise une transformation entre deux systèmes de réécriture par une opération de réécriture afin d'une part d'obtenir des relations entre les positions des termes sources et cibles de la transformation et d'autre part de ramener l'analyse d'une transformation globale à l'analyse d'une transformation locale sera réutilisée aux chapitres 8 et 9 pour établir un lien entre la réduction faible et la réécriture du premier ordre. Ce que nous venons de faire dans la présente section peut être vu comme un exercice d'échauffement en vue de ces deux prochains chapitres.

7.4 Propriétés de contribution

Cette section vérifie que l'étiquetage de la section précédente associé à l'expansion d'un système capture convenablement les créations de radicaux. Nous y verrons également que dans un système non orthogonal, les autres propriétés causales peuvent être plus problématiques.

7.4.1 Contribution directe

Considérons un CRS Σ . Prenons son expansion $E(\Sigma)$ et le raffinement canonique $R(E(\Sigma))$. Appliquons la discipline d'étiquetage définie à la section 7.2 et notons $\Sigma^{\mathcal{L}}$ le système obtenu. Nous pouvons maintenant prouver que $\Sigma^{\mathcal{L}}$ vérifie un lemme de contribution directe identique aux lemmes 5.22 sur le λ -calcul faible étiqueté et 6.59 sur le *Pure Pattern Calculus* faible étiqueté.

Lemme 7.40 (Contribution directe). *Pour tout pas de réduction $\rho : t \rightarrow t'$ de nom Ω , si ρ_c est un pas de réduction de source t' et de nom Ω_c créé par ρ , alors $\Omega \hookrightarrow^d \Omega_c$.*

Démonstration. Notons $\rho : c[l_\alpha^\sigma] \rightarrow c[[\Omega, i(r)]^\sigma]$ le pas de réduction et $l_c^{\sigma_c}$ le radical du pas de réduction créé. En particulier $\Omega_c = \|l_c\|$. Raisononnons par cas sur la création des pas de réduction, avec le lemme 7.7, et montrons que Ω contribue à au moins une position propre de l_c :

- Cas où une position propre p' de $[\Omega, i(r)]^\sigma$ fait partie des positions propres de l_c . Par définition, cette position résulte d'une position propre p de r , et $\tau_{p'}([\Omega, i(r)]^\sigma) = [\Omega, p]$. (nous pouvons aussi remarquer dans ce cas que $p = p'$ en raison du raffinement canonique)
- Cas où la racine de $[\Omega, i(r)]^\sigma$ est une position propre de l_c . Rappelons que nous considérons un système expansé. La racine de r est donc le symbole factice ι , et la racine de $[\Omega, i(r)]^\sigma$ porte alors l'étiquette $[\Omega, \epsilon]$.
- Cas où une position dynamique p' de $[\Omega, r]^\sigma$ fait partie des positions propres de l_c . Le méta-terme r étant issu d'un raffinement canonique, toutes ses méta-variables ont une arité nulle. En particulier $[\Omega, r]^\sigma$ n'a pas de positions dynamiques. \square

Remarque 7.41. *Le troisième cas de création d'un pas de réduction, concernant l'intersection des positions propres du radical du pas de réduction créé avec les positions dynamiques du réduit du pas de réduction effectué, est masqué par l'opération de raffinement canonique. En effet, le raffinement canonique d'un méta-terme peut être vu comme la transformation des positions dynamiques en positions propres.*

7.4.2 Instabilité des radicaux

Au chapitre 6 nous avons défini un système étiqueté non-séquentiel possédant d'autres propriétés causales que le simple témoignage des créations de pas de réduction. Le *Pure Pattern Calculus* étiqueté vérifie en effet le lemme 6.60 de stabilité des radicaux, qui exprime une certaine monotonie de l'évolution des noms de radicaux. L'idée principale en était : l'ensemble des contributeurs d'un résidu contient l'ensemble des contributeurs de son ancêtre.

Une telle propriété ne résiste pas aux cas de non-orthogonalité plus généraux qui se peuvent trouver dans les *CRS*, ou même simplement dans les *TRS* (l'exemple 7.42 suivant est exprimé au premier ordre).

Exemple 7.42.

Considérons des symboles f et a d'arités respectives 2 et 0, et deux règles de réécriture

$$\begin{aligned} \kappa_a & : f(a, Z) \rightarrow f(Z, Z) \\ \kappa_f & : f(Z_1, Z_2) \rightarrow f(Z_2, Z_1) \end{aligned}$$

Plaçons-nous dans l'étiquetage de ce CRS, sans nous soucier de l'expansion qui n'intervient pas ici, et prenons le terme

$$f^\alpha(a^\beta, a^\gamma)$$

De ce terme nous avons un pas de réduction de nom $a\beta$ par la règle κ_f

$$\rho_f : f^\alpha(a^\beta, a^\gamma) \rightarrow f^{[\alpha\beta, \epsilon]}(a^\gamma, a^\gamma)$$

Mais un pas de réduction de nom α par la règle κ_α est également possible

$$\rho_\alpha : f^\alpha(a^\beta, a^\gamma) \rightarrow f^{[\alpha, \epsilon]}(a^\gamma, a^\beta)$$

Maintenant, en considérant le résidu de ρ_f après ρ_α , nous observons un pas de réduction de nom $[\alpha, \epsilon]\gamma$ pour la règle κ_f

$$\rho'_f : f^{[\alpha, \epsilon]}(a^\gamma, a^\beta) \rightarrow f^{[[\alpha, \epsilon]\gamma, \epsilon]}(a^\beta, a^\beta)$$

Visiblement les deux pas de réduction ρ_f et ρ_α de l'exemple 7.42 ne commutent pas, ce qui n'est pas surprenant dans le cadre d'un système non-orthogonal, mais la partie plus problématique est la modification du nom du premier pas de réduction ρ_f par résiduation (ρ'_f) : nous passons de a^β à $[\alpha, \epsilon]\gamma$. Dans le deuxième nom l'étiquette α est toujours présente et apporte ses contributeurs, mais l'étiquette β en revanche a disparu au profit d'une autre étiquette γ qui ne lui est aucunement reliée.

Remarquons que, dans le cadre de la définition 7.1, le pas de réduction ρ'_f n'est pas un résidu exact de pas de réduction ρ_f . En effet, ρ_f utilise la règle étiquetée

$$\kappa_f^l : f^\alpha(a^\beta, Z) \rightarrow f^{[\alpha, \beta, \epsilon]}(Z, Z)$$

tandis que ρ'_f utilise la règle étiquetée

$$\kappa_f^{l'} : f^{[\alpha, \epsilon]}(a^\gamma, Z) \rightarrow f^{[[\alpha, \epsilon]\gamma, \epsilon]}(Z, Z)$$

Ces deux règles étant différentes, la relation entre ρ_f et ρ'_f n'est pas une relation de résiduation exacte. Cependant, les deux règles κ_f^l et $\kappa_f^{l'}$ sont deux étiquetages de la même règle $\kappa_f : f(a, Z) \rightarrow f(Z, Z)$ du système d'origine. Ainsi, dans une version non étiquetée de la même séquence de réduction le pas de réduction ρ'_f serait bien considéré comme un résidu du pas de réduction ρ_f .

Ceci suggère une notion de *résidu approché*, qui désigne des pas de réduction qui sont liés par une relation de résiduation dans le CRS d'origine, mais qui ne sont plus des résidus exacts dans le système expansé, raffiné et étiqueté en raison des informations supplémentaires que contient ce dernier. Pour être plus légitime, une réflexion générale sur la causalité dans les CRS dans un formalisme comme le nôtre devrait se baser sur cette notion de résiduation approchée plutôt que sur la résiduation exacte.

Dans cette thèse cependant, en raison des problèmes d'instabilité des noms évoqués par l'exemple 7.42, nous limiterons la suite de l'étude des propriétés causales de notre étiquetage au cadre des CRS orthogonaux. Dans ce cadre, la résiduation approchée sera toujours exacte, et ces considérations pourront donc être ignorées.

7.5 Correction et complétude causales dans les systèmes orthogonaux

Prenons maintenant un CRS orthogonal faible Σ , et considérons son étiquetage causal avec raffinement et expansion $\Sigma^{\mathcal{L}}$ défini comme ci-dessus. Nous pouvons obtenir dans ce cadre les propriétés de correction et de complétude causales qui avaient été énoncées par les théorèmes 5.27 et 5.29 pour le λ -calcul.

Constatons d'abord que la notion d'antécédent nécessaire peut être adaptée de façon transparente au cadre des CRS faibles orthogonaux.

Définition 7.43 (Ensemble de pas de réduction nécessaire). *Soit \mathcal{T} un ensemble de termes étiquetés. Un ensemble \mathcal{R} de pas de réduction est **nécessaire** pour \mathcal{T} si toute séquence de réduction ρ telle que $\text{src}(\rho)$ est un terme étiqueté initial et $\text{tgt}(\rho) \in \mathcal{T}$ contient un pas de réduction de \mathcal{R} .*

Définition 7.44 (Nom nécessaire). *Soit \mathcal{T} un ensemble de termes étiquetés. Un nom Ω est **nécessaire** pour \mathcal{T} si l'ensemble des pas de réduction de nom Ω est nécessaire pour \mathcal{T} .*

Définition 7.45 (Antécédent nécessaire). *Soient Ω_1 et Ω_2 deux noms de pas de réduction. Le nom Ω_1 est une **antécédent nécessaire** du nom Ω_2 , noté $\Omega_1 \hookrightarrow^n \Omega_2$, si le nom Ω_1 est nécessaire pour l'ensemble des sources de pas de réduction de nom Ω_2 .*

La correction et la complétude causales correspondent toujours à l'égalité des relations \hookrightarrow^n (définition 7.45) et \hookrightarrow (définition 7.24), et les mêmes résultats intermédiaires permettent d'accéder à cette conclusion. La suite de cette section retranscrit le travail qui avait été effectué sur l'étiquetage causal du λ -calcul faible à la section 5.4. Le lemme de contribution directe 5.22 a déjà été généralisé par le lemme 7.40.

Lemme 7.46 (Stabilité des pas de réduction). *Soit un pas de réduction $\rho : t \rightarrow t'$. Si ρ_α est un pas de réduction de source t et de nom Ω , alors tout descendant de $\text{root}(\rho_\alpha)$ est la position racine d'un pas de réduction de même nom Ω .*

Démonstration. Par lemme 4.34, l'étiquetage causal $\Sigma^{\mathcal{L}}$ du CRS orthogonal d'origine est un CRS orthogonal en réduction faible. En particulier, de même que dans un système orthogonal, tout descendant de $\text{root}(\rho_\alpha)$ est la position d'un résidu exact de ρ_α , c'est-à-dire un pas de réduction utilisant la même règle étiquetée $l \rightarrow r$ que ρ_α . Les deux ont donc même nom. \square

Lemme 7.47 (Développements finis). *Soient t un terme étiqueté et \mathcal{R} un ensemble de pas de réduction de source t . Alors les trois propriétés suivantes sont vérifiées :*

- Tous les développements de \mathcal{R} sont finis.
- Tous les développements complets de \mathcal{R} ont même cible.
- Tous les développements complets de \mathcal{R} définissent les mêmes relations de descendance et de résiduation.

Démonstration. Par lemme 4.34, le système étiqueté est un CRS orthogonal en réduction faible. Par conséquent, pour tous deux pas de réduction co-initiaux $t = c_o[l_o^{\sigma_o}] \rightarrow c_o[r_o^{\sigma_o}]$ et $t = c_i[l_i^{\sigma_i}] \rightarrow c_i[r_i^{\sigma_i}]$, si le radical $l_i^{\sigma_i}$ est dans le radical $l_o^{\sigma_o}$ alors $l_i^{\sigma_i}$ est soit dans σ_i soit à une position gelée de l_o . Or $l_i^{\sigma_i}$ n'est pas gelé par hypothèse, donc $l_i^{\sigma_i}$ est dans σ_o . Ainsi, de même que dans un système orthogonal, deux pas de réduction co-initiaux ne se recouvrent jamais. Nous démontrons donc qu'un tel système vérifie les propriétés de développements finis de la même manière que pour un système orthogonal [GK96]. \square

Prouvons d'abord la *correction causale*, c'est-à-dire l'inclusion de la relation de contribution \hookrightarrow dans la relation de dépendance \hookrightarrow^n (théorème 7.50). Le cheminement est le même que celui qui avait été suivi pour le théorème 5.27 de correction causale du λ -calcul faible étiqueté : nous allons suivre les origines des étiquettes et vérifier qu'elles n'enregistrent pas de « fausses » contributions.

Lemme 7.48 (Origine des étiquettes). *Soit ρ une séquence de réduction dont la source est un terme étiqueté initial. Si une étiquette de la forme $[\Omega, \alpha]$ apparaît dans la cible de ρ , alors la séquence ρ contient un pas de réduction de nom Ω .*

Démonstration. Par récurrence sur la longueur de ρ , en remarquant qu'à chaque pas une étiquette de la forme $[\Omega, \alpha]$ soit est créée par un pas de réduction de nom Ω soit est portée par un symbole $f^{[\Omega, \alpha]}$ qui descend d'un symbole identique. \square

Lemme 7.49 (Nécessité directe). *Soit ρ une séquence de réduction dont la source est un terme étiqueté initial. Notons ρ un pas de réduction de source $t = \text{tgt}(\rho)$, et Ω le nom de ρ . Si Ω' est un nom de pas de réduction tel que $\Omega' \hookrightarrow^d \Omega$, alors ρ contient un pas de réduction de nom Ω' .*

Démonstration. Notons $\Omega = \omega_1 \dots \omega_n$. Par définition de \hookrightarrow^d il existe un i tel que $\omega_i = [\Omega', \alpha]$. Donc par définition du nom d'un pas de réduction il existe une position $p \in \text{pos}(t)$ telle que $\tau_p(t) = [\Omega', \alpha]$. Alors par lemme 7.48 la séquence ρ contient un pas de réduction de nom Ω' . \square

Théorème 7.50 (Correction causale). *Soient deux noms de pas de réduction Ω_1 et Ω_2 tels que $\Omega_1 \hookrightarrow \Omega_2$. Alors $\Omega_1 \hookrightarrow^n \Omega_2$.*

Démonstration. Par récurrence sur la longueur d'une séquence $\Omega_1 \hookrightarrow^d \dots \hookrightarrow^d \Omega_2$.

- Si $\Omega_1 \hookrightarrow^d \Omega_2$, alors par lemme 7.49 toute séquence de réduction allant d'un terme initial à un terme source d'un pas de réduction de nom Ω_2 contient un pas de réduction de nom Ω_1 , c'est-à-dire $\Omega_1 \hookrightarrow^n \Omega_2$.
- Si $\Omega_3 \hookrightarrow^n \Omega_2$ et $\Omega_1 \hookrightarrow^d \Omega_3$, alors de même qu'au point précédent $\Omega_1 \hookrightarrow^n \Omega_3$, et par transitivité $\Omega_1 \hookrightarrow^n \Omega_2$. \square

Prouvons enfin la *complétude causale*, c'est-à-dire l'inclusion de la relation de dépendance \hookrightarrow^n dans la relation de contribution \hookrightarrow (théorème 7.52). Le cheminement est le même que celui qui avait été suivi pour le théorème 5.29 de complétude causale du λ -calcul faible étiqueté : l'étape intermédiaire consiste à

vérifier que des étapes de réduction dont les noms ne sont pas comparables par \hookrightarrow peuvent être permutés.

Lemme 7.51 (Permutation). *Si $t_0 \Rightarrow_{\Omega_0} t_1 \Rightarrow_{\Omega} t_2$ et $\Omega_0 \not\prec \Omega$, alors l'un de ces deux cas est vérifié :*

- $t_0 \Rightarrow_{\Omega} t_2$.
- Il existe t'_1 tel que $t_0 \Rightarrow_{\Omega} t'_1 \Rightarrow_{\Omega_0} t_2$.

Démonstration. Comme $\Omega_0 \not\prec \Omega$, par contraposition du lemme 7.40 les pas de réduction de source t_1 et de nom Ω ont des ancêtres de source t_0 . En particulier $t_0 \Rightarrow_{\Omega_0} t_1 \Rightarrow_{\Omega} t_2$ est un développement complet de l'ensemble des pas de réduction de source t_1 et de nom Ω_0 ou Ω et par lemme 7.47 sa cible t_2 est également la cible de tout autre développement complet, par exemple le développement consistant d'abord en la réduction parallèle du nom Ω suivie de celle du nom Ω_0 si ce dernier apparaît encore. \square

Théorème 7.52 (Complétude causale). *Soient deux noms de pas de réduction Ω_0 et Ω tels que $\Omega_0 \hookrightarrow^n \Omega$. Alors $\Omega_0 \hookrightarrow \Omega$.*

Démonstration. Considérons un nom de pas de réduction Ω , et $\bar{\mathcal{R}}$ l'ensemble des séquences de réductions parallèles étiquetées de longueur minimale ayant pour source un terme initial et terminant par une étape Ω . Il existe donc un n tel que toute séquence $\bar{\rho} \in \bar{\mathcal{R}}$ a la forme

$$\Rightarrow_{\Omega_1^e} \dots \Rightarrow_{\Omega_n^e} \Rightarrow_{\Omega}$$

Notons $\bar{\mathcal{R}}^*$ l'ensemble des séquences $\bar{\rho} \in \bar{\mathcal{R}}$ pour lesquelles il existe au moins un $i \in \{1, \dots, n\}$ tel que $\Omega_i^e \not\prec \Omega$.

Supposons que l'ensemble $\bar{\mathcal{R}}^*$ est non vide. Notons i le plus grand entier tel qu'il existe une séquence $\bar{\rho} \in \bar{\mathcal{R}}^*$ telle que $\Omega_i^e \not\prec \Omega$. Notons $\bar{\rho}$ une séquence de $\bar{\mathcal{R}}^*$ telle que $\Omega_i^e \not\prec \Omega$. La séquence $\bar{\rho}$ est de la forme

$$\dots \Rightarrow_{\Omega_i^e} \Rightarrow_{\Omega_{i+1}^e} \dots \Rightarrow_{\Omega_n^e} \Rightarrow_{\Omega}$$

(où l'étape $\Rightarrow_{\Omega_{i+1}^e}$ peut être confondue avec l'étape $\Rightarrow_{\Omega_n^e}$ ou l'étape \Rightarrow_{Ω}).

Par définition de $\bar{\rho}$, nous avons $\Omega_{i+1}^e \hookrightarrow \Omega$. Comme $\Omega_i^e \not\prec \Omega$ et par transitivité de \hookrightarrow nous en déduisons $\Omega_i^e \not\prec \Omega_{i+1}^e$. Alors par lemme 7.51 nous pouvons construire l'une des deux séquences suivantes :

- Une séquence $\dots \Rightarrow_{\Omega_{i-1}^e} \Rightarrow_{\Omega_{i+1}^e} \dots \Rightarrow_{\Omega}$ strictement plus courte que $\bar{\rho}$, contredisant la minimalité de sa longueur parmi les séquences de réductions parallèles étiquetées s'achevant par \Rightarrow_{Ω} .
- Une séquence $\dots \Rightarrow_{\Omega_{i+1}^e} \Rightarrow_{\Omega_i^e} \dots \Rightarrow_{\Omega}$ avec $\Omega_i^e \not\prec \Omega$, de même longueur que $\bar{\rho}$ mais contredisant la maximalité de i .

Ainsi, l'ensemble $\bar{\mathcal{R}}^*$ est nécessairement vide, et tout nom Ω_0 apparaissant dans une séquence de $\bar{\mathcal{R}}$ vérifie $\Omega_0 \hookrightarrow \Omega$.

Finalement, soit Ω_0 un nom tel que $\Omega_0 \hookrightarrow^n \Omega$. Par définition 7.45, toute séquence de réduction ρ dont la source est initiale et dont la cible $\text{tgt}(\rho)$ est

source d'un pas de réduction ρ de nom Ω contient un pas de nom Ω_0 . C'est le cas en particulier de toute séquence développant une séquence de $\bar{\mathcal{R}}$. Donc $\Omega_0 \hookrightarrow \Omega$. \square

Ainsi, si comme remarqué à la section 7.4 l'étiquetage « causal » des *CRS* n'a que peu de propriétés causales dans le cas général, il a en revanche bien nos deux propriétés de correction et de complétude causales dans le cas de *CRS* orthogonaux (théorèmes 7.50 et 7.52), et la preuve de ces propriétés n'est pas plus difficile que dans le cadre du λ -calcul (théorèmes 5.27 et 5.29). Ces propriétés justifient aussi *a posteriori* la légitimité de l'utilisation du raffinement canonique : bien que le raffinement fasse « gonfler » les noms des pas de réductions l'opération n'a pas d'impact sur les relations de contribution.

Bilan du chapitre

Dans ce chapitre nous avons généralisé aux *CRS* faibles l'analyse de causalité faite pour le λ -calcul faible au chapitre 5 et le *Pure Pattern Calculus* faible au chapitre 6 (section 7.1). Une fois encore, cette analyse de causalité a abouti à un étiquetage des *CRS* (section 7.2), dont nous pouvons noter qu'il est significativement plus simple que l'étiquetage du *Pure Pattern Calculus* du chapitre 6 grâce au formalisme des *CRS* associé à la notion de raffinement canonique héritée du chapitre 4.

Nous avons vu que ces *CRS* étiquetés n'ont pas en général toutes les propriétés causales qu'a le λ -calcul faible étiqueté, mais que ces propriétés de correction et de complétude causales sont retrouvées dans le cas de *CRS* orthogonaux (section 7.5). Nous pouvons donc en déduire que les *CRS* orthogonaux faibles sont des systèmes dans lesquels la dépendance causale est bien définie, ce qui nous permettra dans l'ultime partie de cette thèse (au chapitre 10) d'y parler de partage optimal.

Enfin, nous avons établi un lien entre l'étiquetage causal des *CRS* faibles et l'étiquetage pleinement paresseux défini au chapitre 4, lien qui nous permettra de faire remonter au cadre pleinement paresseux ces résultats d'optimalité qui seront établis au chapitre 10.

Mais avant cela nous allons faire dans la partie IV suivante un détour simplificateur et analyser les relations entre la réduction faible et la réécriture du premier ordre. Nous allons pour cela étudier une transformation de systèmes de réécriture en suivant les mêmes méthodes qui ont été introduites à la section 7.3.3 pour l'analyse de l'expansion/compression d'un système, à savoir la décomposition d'une transformation globale en une transformation « à petits pas » et son intégration dans un système de réécriture contenant en outre à la fois le système source et le système cible de la transformation.

QUATRIÈME PARTIE

PREMIER ORDRE

Introduction

La présente partie a pour objectif la formalisation d'une idée qui est dans l'air depuis longtemps dans le monde de la programmation fonctionnelle comme dans celui de la réécriture : la réduction faible peut être ramenée à de la réécriture du premier ordre. Ce fait a déjà pu se matérialiser par des traductions du λ -calcul en logique combinatoire [CF74, cH98] ou dans des systèmes de super-combinateurs [Hug82, Joh85, PJ87], dans des systèmes de substitutions explicites [ACCL90, Kes07], dans des transformations de défonctionalisation [Rey98], ou encore dans la dérivation de machines abstraites [Kri07, Ses97, ABDM03].

La correspondance obtenue entre ces deux mondes va toutefois plus loin que ce qui était suggéré jusque là, puisqu'elle cumule toutes les caractéristiques suivantes :

- elle donne une bisimilarité entre tout système faible et sa traduction au premier ordre,
- elle préserve la descendance et la résiduation,
- elle préserve l'éventuelle orthogonalité des systèmes, et
- elle préserve le partage.

Ainsi au terme de cette partie nous obtiendrons un lien fort entre le partage pleinement paresseux dans un *CRS* faible arbitraire et une certaine réduction partagée dans un système du premier ordre associé, de telle manière que nous pourrions conclure dans la partie suivante à l'optimalité de la pleine paresse grâce à la seule preuve (bien plus simple) de l'optimalité de la réduction partagée dudit système du premier ordre.

Le chapitre 8 introduit la technique utilisée dans le cas du λ -calcul, d'où elle tire son origine. La correspondance entre réduction faible et réécriture du premier ordre va en effet être obtenue par une certaine formalisation de la technique de compilation bien connue qu'est le λ -*lifting* [Joh85]. Comme à la section 7.3.3 pour l'opération de compression, la transformation sera formalisée « à petits pas » : non comme une opération globale mais comme une séquence d'opérations locales, en explicitant les étapes intermédiaires de la transformation. Ceci alourdira le formalisme mais simplifiera le raisonnement. Au passage, ce lien permettra aussi de compléter l'analyse du chapitre 3 en formalisant l'équivalence entre le λ -*lifting* pleinement paresseux [Hug82, PJ87] et les caractérisations déjà étudiées de la pleine paresse.

Le chapitre 9 généralise l'opération au cadre des *CRS*. Ce cadre plus riche demande un investissement de départ supérieur au moment des définitions de base des opérations de transformation. En revanche, les *CRS* sont un cadre naturel dans lequel exprimer tout à la fois le *CRS* source et le système du premier ordre cible, ainsi que toutes les étapes intermédiaires de la transformation, et c'est cette version plus générale qui en définitive sera aussi la plus simple et la plus concise.

CHAPITRE 8

Lambda-lifting pleinement paresseux

Sommaire

8.1 Extractions	252
8.2 <i>Lambda-lifting</i>	254
8.3 Bisimulation	264
8.4 Discussion : réduction faible et orthogonalité	268
8.5 <i>Lambda-lifting</i> étiqueté	269

L'analyse de la causalité dans le λ -calcul menée au chapitre 5 nous a permis de constater des différences significatives entre les comportements de la β -réduction générale et de la β -réduction faible. Nous allons maintenant donner une explication à ces différences constatées, en explicitant la nature « premier ordre » de la réduction faible.

Pour ce faire, le présent chapitre donne une nouvelle présentation du *λ -lifting*, une transformation de programmes bien connue [Joh85, PJ87] qui permet de passer d'un programme fonctionnel (un λ -terme) à une expression applicative (un terme du premier ordre). Cette présentation met en œuvre à plus grande échelle les principes illustrés à la section 7.3.3 pour l'étude de la transformation de compression d'un système : la transformation globale de *λ -lifting* va être représentée « à petits pas » dans le cadre d'un système de réécriture qui permettra de raisonner sur les étapes intermédiaires de la transformation. Ainsi nous pourrons prouver par un raisonnement de permutation locale de pas de réductions que notre *λ -lifting* définit une bisimulation entre le λ -calcul faible et un système de réécriture du premier ordre orthogonal. Mieux, mais nous ne le formaliserons qu'au chapitre 9 suivant, cette bisimulation est compatible avec le partage pleinement paresseux (ce qui permet d'assimiler le *λ -lifting* pleinement paresseux à la notion unifiée de pleine paresse formalisée au chapitre 3), ainsi qu'avec les notions de descendance, de résiduation, et de nécessité.

La section 8.1 rappelle les principes du *λ -lifting* et du *λ -lifting* pleinement paresseux tels qu'ils peuvent être trouvés dans la littérature, puis la section 8.2

en présente notre formalisation « à petits pas » dans un cadre de réécriture et la section 8.3 établit la bisimilarité entre le λ -calcul faible et le système du premier ordre orthogonal qui est la cible de la transformation de λ -lifting (nous obtiendrons donc une nouvelle preuve de la correction de cette transformation). Le rôle de la réduction faible dans cet assemblage est discuté à la section 8.4. La section 8.5 considère enfin une version étiquetée du λ -lifting qui définit un SLS.

8.1 Extractions

L'idée directrice du λ -lifting est, à partir d'un programme fonctionnel d'ordre supérieur, de remplacer les fonctions par des objets plus simples appelés *supercombinateurs*. Conjointement à ce processus, des mécanismes d'évaluation sont ajoutés pour que les supercombinateurs se comportent de la même façon que les fonctions qu'ils remplacent.

Exemple 8.1.

Considérons ce code qui, dans un style à la ML, applique une fonction anonyme à un argument :

```
| (fun x -> x + 2) 1
```

Le λ -lifting remplace la fonction anonyme par un supercombinateur \$F\$, ce qui donne le code transformé

```
| $F 1
```

et génère l'équation

```
| $F x = x + 2
```

Ainsi le code transformé peut s'évaluer en 3.

Une des caractéristiques d'un supercombinateur est qu'il représente un programme clos. Cela signifie que le code correspondant peut être compilé une fois pour toutes, et séparément du reste du programme. Cet aspect clos n'est pas naturel avec les fonctions d'ordre supérieur, qui peuvent contenir des *éléments libres*. La transformation en supercombinateur d'une telle fonction requiert ainsi l'extraction de ses éléments libres. En quelque sorte, toute fonction doit être explicitement associée à l'ensemble des paramètres extérieurs auxquels elle fait référence.

Exemple 8.2.

Considérons ce nouveau code, où cette fois la fonction appliquée contient une variable libre y :

```
| (fun x -> x + (y * y)) 1
```

Le supercombinateur destiné à remplacer la fonction doit cette fois prendre la variable y en paramètre. Nous obtenons le code

```
| $G y 1
```

où le supercombinateur $\$G$ est défini par l'équation

```
| $G y x = x + (y * y)
```

Ainsi, quelle que soit la valeur y associée à y le code transformé peut s'évaluer en $1 + y^2$.

Il y a ensuite une certaine latitude dans l'interprétation de ce point. Les « éléments libres » peuvent en effet être simplement les variables libres de la fonction, ainsi qu'il est fait dans le λ -*lifting* le plus traditionnel [Joh85, PJ87]. Mais ils peuvent plus généralement être n'importe quelle expression libre de la fonction. Le λ -*lifting* pleinement paresseux [Hug82, PJ87] en particulier extrait d'une fonction ses expressions libres maximales.

Exemple 8.3.

Considérons encore le code :

```
| (fun x -> x + (y * y)) 1
```

L'expression $y * y$ est indépendante de la variable x de la fonction, et est donc une expression libre, qui peut être intégralement extraite comme un seul paramètre. Nous obtenons cette fois un code transformé

```
| $H (y * y) 1
```

et la définition d'un supercombinateur

```
| $H z x = x + z
```

dont la première variable est supposée recevoir l'expression libre qui a été extraite et la deuxième l'argument d'origine de la fonction représentée.

Nous utiliserons dans ce chapitre cette discipline de λ -*lifting* pleinement paresseux, qui s'articule avec le reste des discussions d'une manière particulièrement agréable. Dans cette présentation du λ -*lifting*, qui est celle utilisée par Peyton-Jones [PJ87], les supercombinateurs sont des constantes qui s'appliquent de manière curryfiée à une série de termes jouant deux rôles : les premiers termes sont les paramètres libres extraits de la fonction (à l'exemple 8.3, variable z pour paramètre libre $y * y$), et les deuxièmes sont les arguments de la fonction tels qu'indiqués par le code originel (à l'exemple 8.3, variable d'origine 1 pour argument d'origine 1).

Pour rester plus proches de la structure du terme d'origine, nous allons marquer la différence entre ces deux sortes d'arguments/paramètres. Dans la suite un supercombinateur sera un symbole dont l'arité correspond au nombre de paramètres extraits du corps de la fonction associée. Ainsi les paramètres extraits apparaîtront comme les fils du supercombinateur dans le terme obtenu, tandis que les arguments indiqués par le code originel conserveront leur place à part, associée à une application (voir exemple 8.4).

Avec cette nouvelle manière de présenter le λ -*lifting*, le code produit par la transformation d'un programme prend la forme d'un terme sur une signature mixte formée d'une part des constructions de base comme l'application et d'autre part de l'ensemble des supercombinateurs utilisés. Les équations de définition des supercombinateurs prennent alors la forme de règles de réécriture du premier ordre, ainsi que cela apparaît dans une transformation proposée par Dal Lago et Martini [LM09b].

Exemple 8.4.

Le code

```
| (fun x -> x + (y * y)) 1
```

peut donc être transformé en un terme

$$@ (h(y*y), 1)$$

où le supercombinateur est représenté par un symbole h d'arité 1 associé à la règle de réécriture

$$@ (h(Z), X) \rightarrow X + Z$$

8.2 Lambda-lifting

Appliquer la transformation classique de λ -*lifting* à un programme donné ne nécessite qu'un nombre fini de supercombinateurs, c'est-à-dire un par expression fonctionnelle dans le programme, qui sont créés à la volée. Dans ce chapitre cependant nous allons regarder la traduction de tout le λ -calcul et pas seulement d'un unique terme. Pour cela nous aurons besoin d'une infinité de symboles et d'une fonction fixant la correspondance entre les symboles de supercombinateurs et les fragments de termes associés.

Convention 8.5. *Fixons pour tout $n \in \mathbb{N}$ un ensemble dénombrable \mathcal{F}_n de symboles d'arité n , et définissons $\mathcal{F} = \bigcup_n \mathcal{F}_n$. Les supercombinateurs seront représentés par les symboles de l'ensemble \mathcal{F} . Plus précisément, un symbole de \mathcal{F}_n représentera un supercombinateur dont n éléments libres ont été extraits.*

Nous introduirons bientôt une fonction d'*expansion* qui matérialisera cette représentation en associant à chaque symbole d'arité n un contexte d'arité n sur une signature bien choisie.

Rappelons également que nous allons donner une formalisation « à petits pas » du λ -*lifting*. Plus précisément, chaque pas de λ -*lifting* correspondra à la transformation d'une fonction. Ceci nous demande de pouvoir décrire toutes les étapes intermédiaires de la transformation. Nous nous plaçons donc dans ce chapitre dans une syntaxe mixte regroupant les constructions habituelles du λ -calcul avec ces nouveaux symboles de supercombinateurs.

Définition 8.6 (Syntaxe étendue du λ -calcul). *Les λ -termes étendus sont définis par la syntaxe étendue suivante :*

$$t = x \mid \lambda x.t \mid tt \mid f(t_1, \dots, t_n) \quad f \text{ d'arité } n$$

Les notions de contextes, positions, variables libres, et substitutions des sections 0.3 et 0.5 sont directement étendues à cette syntaxe.

Comme nous considérons ici le λ -lifting pleinement paresseux, qui extrait des fonctions leurs expressions libres maximales avant de les convertir en supercombinateurs, les fragments de termes correspondant aux supercombinateurs sont les squelettes qui peuvent être construits sur la syntaxe étendue.

Définition 8.7 (Squelettes). *La définition inductive des θ -squelettes est étendue par un cas hérité des CRS. Rappelons que le nombre d'occurrences du trou \square définit l'arité d'un squelette.*

$$\begin{aligned} \langle\langle t \rangle\rangle^\theta &= \square & \theta \cap \text{fv}(t) &= \emptyset \\ \text{Sinon :} & & & \\ \langle\langle x \rangle\rangle^\theta &= x & x &\in \theta \\ \langle\langle \lambda x.t \rangle\rangle^\theta &= \lambda x. \langle\langle t \rangle\rangle^{\theta \cup \{x\}} \\ \langle\langle t_1 t_2 \rangle\rangle^\theta &= \langle\langle t_1 \rangle\rangle^\theta \langle\langle t_2 \rangle\rangle^\theta \\ \langle\langle f(t_1, \dots, t_n) \rangle\rangle^\theta &= f(\langle\langle t_1 \rangle\rangle^\theta, \dots, \langle\langle t_n \rangle\rangle^\theta) \end{aligned}$$

Notons $\langle\langle \Lambda \rangle\rangle$ l'ensemble des squelettes de toutes les λ -abstractions sur la syntaxe étendue.

$$\langle\langle \Lambda \rangle\rangle = \{ \lambda x. \langle\langle t \rangle\rangle^{\{x\}} \mid x \in \mathcal{X}, t \text{ un } \lambda\text{-terme étendu} \}$$

Définition 8.8 (Lambda-expandeur). *Un λ -expandeur est une fonction bijective $\psi : \mathcal{F} \rightarrow \langle\langle \Lambda \rangle\rangle$ qui préserve l'arité. Pour tous $f, g \in \mathcal{F}$ nous disons que g **contient** f et nous notons $g \supset f$ quand f apparaît dans l'expansion $\psi(g)$ de g . Le λ -expandeur ψ est **bien fondé** quand il n'existe pas de suite infinie décroissante $f_1 \supset f_2 \supset \dots$*

Notons que cette notion d'expandeur n'a rien de commun avec l'expansion d'un système de réécriture vue à la section 7.2.

Exemple 8.9.

Considérons trois symboles f, g et h d'arité 2. Soit une fonction $\psi : \mathcal{F} \rightarrow \langle\langle \Lambda \rangle\rangle$ telle que

$$\begin{aligned} \psi(f) &= \lambda x. \square g(\square, x) \\ \psi(g) &= \lambda y. h(y, \square) \square \end{aligned}$$

Nous avons en particulier $f \supset g$ et $g \supset h$ et rien ne s'oppose à ce que ψ soit un λ -expandeur bien fondé. Si en revanche nous ajoutons la définition

$$\psi(h) = \lambda z. \square f(z, \square)$$

alors nous obtenons de plus $h \supset f$ et nous pouvons former une chaîne infinie décroissante $f \supset g \supset h \supset f \supset \dots$. Alors ψ n'est pas bien fondé.

Convention 8.10. Fixons pour tout ce chapitre un λ -expanseur bien fondé ψ . Remarquons au passage qu'une telle bijection entre \mathcal{F} et $\langle\langle\Lambda\rangle\rangle$ existe car pour tout entier n l'ensemble des contextes à n trous sur la syntaxe étendue est dénombrable.

Étant donné un λ -terme étendu t qui contient des symboles de \mathcal{F} , le λ -expanseur bien fondé ψ décrit une structure qui est implicite dans t , car cachée dans les symboles de \mathcal{F} . Pour rendre partiellement visible cette structure nous définissons les *positions étendues* d'un λ -terme étendu t , qui regroupent les positions usuelles de t et les positions « internes » des symboles de \mathcal{F} .

Définition 8.11 (Positions étendues). L'ensemble des **positions étendues** par ψ d'un λ -terme étendu t , ou ψ -**positions** de t , noté $\text{pos}_\psi(t)$, est défini par :

$$\begin{aligned} \text{pos}_\psi(x) &= \{\epsilon\} \\ \text{pos}_\psi(\lambda x.t) &= \{\epsilon\} \cup 1 \cdot \text{pos}_\psi(t) \\ \text{pos}_\psi(t_1 t_2) &= \{\epsilon\} \cup 1 \cdot \text{pos}_\psi(t_1) \cup 2 \cdot \text{pos}_\psi(t_2) \\ \text{pos}_\psi(f(t_1, \dots, t_n)) &= \{\epsilon\} \cup (0 \cdot \text{pos}_\psi(\psi(f))) \cup (\bigcup_i i \cdot \text{pos}_\psi(t_i)) \quad f \in \mathcal{F} \end{aligned}$$

Observons que cet ensemble est fini car ψ est bien fondé (convention 8.10).

Nous pouvons remarquer dans cette définition que le nombre d'occurrences du chiffre 0 dans une position étendue correspond au nombre de fois où des symboles de \mathcal{F} doivent être expansés pour révéler cette position. Ceci est visible également dans l'exemple 8.12 suivant.

Exemple 8.12.

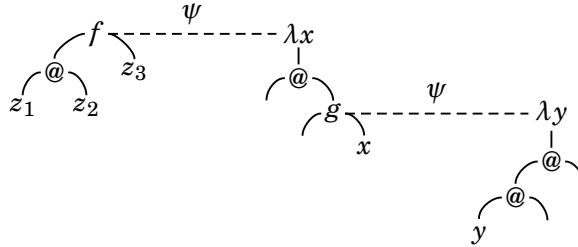
Considérons le λ -terme étendu

$$t = f(z_1 z_2, z_3)$$

ainsi que les expansions suivantes où f et g sont d'arité 2 :

$$\begin{aligned} \psi(f) &= \lambda x. \square g(\square, x) \\ \psi(g) &= \lambda y. y \square \square \end{aligned}$$

Le schéma suivant montre à la fois t et les expansions des symboles f et g , qui sont attachées aux occurrences de ces symboles.



L'ensemble des positions de t est $\text{pos}(t) = \{\epsilon, 1, 11, 12, 2\}$. L'ensemble des positions étendues de t est

$$\text{pos}_\psi(t) = \{\epsilon, 1, 11, 12, 2, 0, 01, 012, 0122, 0120, 01201, 012011, 0120111\}$$

où la table suivante donne une correspondance entre certaines positions étendues et des symboles de t , de $\psi(f)$ ou de $\psi(g)$:

ϵ	f
0	λx
012	g
0120	λy

Définition 8.13 (*Lambda-lifting*). Soit $\lambda x.t$ une λ -abstraction étendue. Notons $t = s[t_1, \dots, t_n]$ avec $s = \langle\langle t \rangle\rangle^{(x)}$ (en rappelant qu'un squelette est un contexte). Nous avons alors le schéma de λ -**lifting** :

$$\lambda x.s[t_1, \dots, t_n] \rightarrow_{\text{ft}} \psi^{-1}(\lambda x.s)(t_1, \dots, t_n)$$

Un pas de λ -lifting est l'application de ce schéma de réduction dans un contexte arbitraire.

Remarquons que les pas de réduction de λ -lifting ne sont pas soumis à la restriction de réduction faible. Comme nous allons le voir à l'exemple 8.14 cette restriction de réduction faible ne serait pas pertinente pour un tel mécanisme de transformation de programmes.

En utilisant ce schéma de λ -lifting, nous pouvons étudier la transformation décrite par Peyton-Jones [PJ87], qui impose comme restriction de « toujours transformer une λ -abstraction la plus profonde ». C'est-à-dire ici de n'appliquer la règle qu'à des λ -abstractions qui ne contiennent pas parmi leurs sous-termes d'autres λ -abstractions.

Exemple 8.14.

Considérons le λ -terme $\lambda x.z_1z_2(\lambda y.yz_3x)$

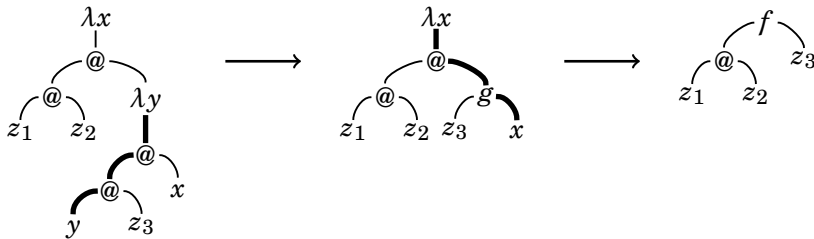
qui présente deux λ -abstractions. La plus profonde est la λ -abstraction $\lambda y.yz_3x$, dont le squelette est $\lambda y.y\square\square$. En notant $g = \psi^{-1}(\lambda y.y\square\square)$ nous pouvons appliquer le schéma de λ -lifting et obtenir le λ -terme étendu

$$\lambda x.z_1z_2g(z_3, x)$$

qui est une λ -abstraction de squelette $\lambda x.\square g(\square, x)$. Notons $f = \psi^{-1}(\lambda x.\square g(\square, x))$. Nous pouvons effectuer un pas de λ -lifting supplémentaire et aboutir au λ -terme étendu

$$f(z_1z_2, z_3)$$

En traçant en gras le squelette de la λ -abstraction à laquelle s'applique chaque pas de λ -lifting nous pouvons représenter graphiquement la précédente séquence par :



Par cette stratégie de transformation donnant la priorité aux λ -abstractions les plus profondes nous pouvons mener au bout le λ -lifting de tout λ -terme en n'appliquant chaque étape de transformation qu'à des λ -abstractions qui ne contiennent pas d'autres λ -abstractions. Cependant certaines descriptions du λ -lifting ne font aucune mention de cette restriction [AF97, Ses97] et semblent utiliser le schéma de réduction général. Remarquons qu'en l'état actuel de notre définition du λ -lifting, ne pas suivre la stratégie de Peyton-Jones détruit la confluence de la transformation.

Exemple 8.15.

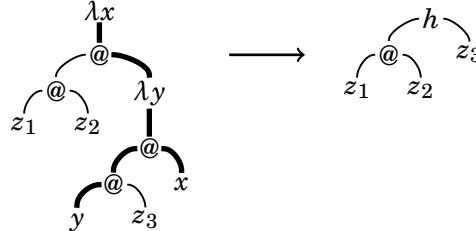
Considérons le λ -terme de l'exemple 8.14 :

$$\lambda x.z_1z_2(\lambda y.yz_3x)$$

Le squelette de sa λ -abstraction racine est $\lambda x.\square(\lambda y.y\square x)$. Notons $h = \psi^{-1}(\lambda x.\square(\lambda y.y\square x))$, nous pouvons appliquer le schéma de réduction du λ -lifting et obtenir le terme étendu

$$h(z_1z_2, z_3)$$

qui est une forme normale pour les pas de λ -lifting différente de $f(z_1z_2, z_3)$. Graphiquement :



Ce phénomène vient du fait que la « forme normale » $h(z_1z_2, z_3)$ représente en fait une transformation incomplète. Le symbole h est en effet associé à un squelette $\lambda x.\square(\lambda y.y\square x)$ qui contient encore une λ -abstraction interne. Ce caractère incomplet n'apparaît ici qu'au niveau *méta* avec le λ -expandeur ψ .

À l'aide de constructions `let...in...` nous pourrions proposer encore une présentation alternative du λ -lifting, dans laquelle le contenu des supercombinateurs ne serait pas caché, et leur transformation apparaîtrait alors naturelle. L'exemple 8.16 illustre ce que pourrait être une telle présentation avec les transformations des exemples 8.14 et 8.15.

Exemple 8.16.

Dans cette présentation informelle, chaque supercombinateur est nommé par une construction `let...in...` qui permet de rendre son « contenu » apparent. La séquence de réduction de l'exemple 8.14 peut être exprimée par :

$$\begin{array}{c}
\lambda x.(z_1 z_2)(\lambda y.y z_3 x) \\
\downarrow \\
\text{let } x_g = \lambda w_1 w_2 y.y w_1 w_2 \\
\text{in } \lambda x.(z_1 z_2)(x_g z_3 x) \\
\downarrow \\
\text{let } x_g = \lambda w_1 w_2 y.y w_1 w_2 \\
\text{in let } x_f = \lambda w_3 w_4 x.w_3(x_g w_4 x) \\
\text{in } x_f(z_1 z_2)z_3
\end{array}$$

Nous pouvons également représenter le pas de réduction de l'exemple 8.15, suivi d'un pas de transformation « à l'intérieur » de h .

$$\begin{array}{c}
\lambda x.(z_1 z_2)(\lambda y.y z_3 x) \\
\downarrow \\
\text{let } x_h = \lambda w_3 w_4 x.w_3(\lambda y.y w_4 x) \\
\text{in } x_h(z_1 z_2)z_3 \\
\downarrow \\
\text{let } x_h = \lambda w_3 w_4 x.w_3 (\text{let } x_g = \lambda w_1 w_2 y.y w_1 w_2 \\
\text{in } x_g w_4 x) \\
\text{in } x_h(z_1 z_2)z_3
\end{array}$$

Le résultat obtenu au terme de cette séquence est équivalent, à réorganisation des constructions $\text{let} \dots \text{in} \dots$ près, au résultat obtenu au terme de la séquence précédente.

Dans un souci de généralité, nous allons donc élargir la définition des pas de λ -lifting de manière à inclure les transformations qui opèrent au niveau *méta*. Ainsi nous couvrirons les approches plus générales et obtiendrons un système de réécriture convergent pour représenter la transformation.

Définition 8.17 (*Méta- λ -lifting*). Les pas de **méta- λ -lifting** sont définis par les conditions suivantes :

Réification. Si $t = s[t_1, \dots, t_n]$ avec $s = \langle\langle t \rangle\rangle^{(x)}$, alors

$$\lambda x.t \xrightarrow{\varepsilon}_{\text{ift}} \psi^{-1}(\lambda x.s)(t_1, \dots, t_n)$$

Contexte. Si $t \xrightarrow{p}_{\text{ift}} t'$ et c est un contexte étendu unaire dont le trou a la position q , alors

$$c[t] \xrightarrow{q \cdot p}_{\text{ift}} c[t']$$

Méta. Si $t = s[t_1, \dots, t_n]$ avec $s = \langle\langle t \rangle\rangle^{(x)}$ et $\lambda x.s[t_1, \dots, t_n] \xrightarrow{1 \cdot p}_{\text{ift}} \lambda x.s'[t_1, \dots, t_n]$, alors

$$\psi^{-1}(\lambda x.s)(t_1, \dots, t_n) \xrightarrow{01 \cdot p}_{\text{ift}} \psi^{-1}(\lambda x.s')(t_1, \dots, t_n)$$

Notons que dans la condition $\lambda x.s[t_1, \dots, t_n] \xrightarrow{1 \cdot p}_{\text{ift}} \lambda x.s'[t_1, \dots, t_n]$ du cas *Méta*, les t_i pourraient être remplacés par n'importe quelle expression libre dans s .

Exemple 8.18.

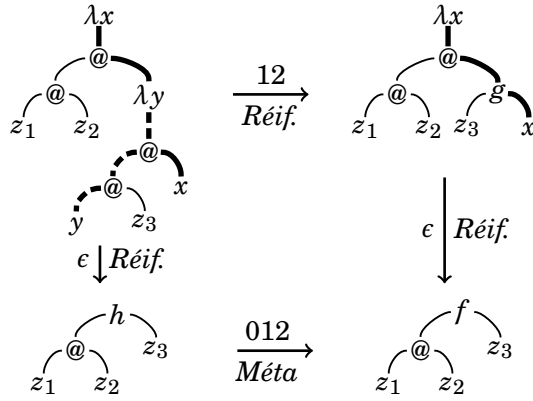
Considérons le λ -terme des exemples 8.14 et 8.15 :

$$\lambda x.z_1 z_2 (\lambda y.y z_3 x)$$

Rappelons les notations :

$$\begin{aligned} g &= \psi^{-1}(\lambda y.y \square \square) \\ f &= \psi^{-1}(\lambda x.\square g(\square, x)) \\ h &= \psi^{-1}(\lambda x.\square(\lambda y.y \square x)) \end{aligned}$$

Les lignes pointillées désignent ci-dessous le squelette de la λ -abstraction λy , et les lignes grasses (pleines ou pointillées) désignent le squelette de la λ -abstraction λx . Avec une étape de méta- λ -lifting nous pouvons maintenant joindre les deux séquences de réduction des exemples 8.14 et 8.15 :



Il sera intéressant pour la suite, à commencer par le lemme 8.25 de convergence qui assurera que les règles de méta- λ -lifting définissent bien une fonction, de remarquer que ce méta- λ -lifting peut également être vu à travers deux schémas de réduction réciproques, qui permettent de « faire » et de « défaire » une opération de λ -lifting pour ramener une réduction Méta à une réification normale.

Définition 8.19 (Contraction/Expansion). La **contraction** et l'**expansion** sont deux schémas de réduction définis par :

$$\begin{aligned} \text{(Contraction)} \quad \lambda x.t &\rightarrow_c \psi^{-1}(\lambda x.s)(t_1, \dots, t_n) & t = s[t_1, \dots, t_n], s = \langle\langle t \rangle\rangle^{(x)} \\ \text{(Expansion)} \quad f(t_1, \dots, t_n) &\rightarrow_e \psi(f)[t_1, \dots, t_n] & \forall i.t_i \text{ libre dans } \psi(f)[t_1, \dots, t_n] \end{aligned}$$

Remarque 8.20. Pour tous deux λ -termes étendus t et u , nous avons $t \rightarrow_c u$ si et seulement si $u \rightarrow_e t$.

Remarque 8.21. Par la condition sur l'expansion, si nous considérons que $\psi(f)$ est un terme syntaxique dont nous ne pouvons renommer les variables, alors certaines occurrences de symboles de supercombinateurs ne peuvent pas être expansées. Une parade évidente à cela consiste à autoriser le renommage des variables de $\psi(f)$ de sorte que la condition soit remplie.

Il est également possible de marginaliser ce problème sans utiliser de renommage : toute occurrence de symbole de supercombinateur introduite par une contraction est sous une forme qui autorise son expansion, et ce caractère est stable à la fois par contraction et par β -réduction faible, cette dernière ne requérant pas de renommage de variables si le λ -terme initial est bien choisi. Ainsi, si nous ne considérons que des λ -termes étendus qui sont obtenus par des séquences de β -réduction faible, contraction, et expansion à partir de λ -termes « purs » dont les variables sont toutes différentes, alors l'expansion est toujours possible.

Quoi qu'il en soit, la forme réelle de cette forme totalement expansée n'importera pas dans la suite.

Tout pas de *méta- λ -lifting* peut être décomposé en une séquence d'expansions et de contractions selon le principe suivant (formalisé par le lemme 8.24) :

- Un pas de *réification* est directement une contraction.
- Un pas *méta* est un pas de *méta- λ -lifting* précédé d'un pas d'expansion et suivi d'un pas de contraction, ces deux derniers étant « inverses » l'un de l'autre (ils ne sont pas rigoureusement inverses puisque la contraction s'applique après modification du terme, mais les deux agissent à la même position).

Exemple 8.22.

Considérons le pas de réduction méta de l'exemple 8.18 :

$$h(z_1 z_2, z_3) \rightarrow_{lft} f(z_1 z_2, z_3)$$

Nous pouvons le décomposer en la séquence d'expansions et de contractions suivante :

$$\begin{array}{lll} h(z_1 z_2, z_3) & & \\ \rightarrow_e \lambda x.z_1 z_2(\lambda y.y z_3 x) & \text{avec} & \psi(h) = \lambda x.\square(\lambda y.y\square x) \\ \rightarrow_c \lambda x.z_1 z_2 g(z_3, x) & \text{avec} & \psi^{-1}(\lambda y.y\square x) = g \\ \rightarrow_c f(z_1 z_2, z_3) & \text{avec} & \psi^{-1}(\lambda x.\square g(\square, x)) = f \end{array}$$

Convention 8.23. Pour toute notion de réduction \rightarrow et tout entier $k \geq 0$, nous notons $t \rightarrow^k t'$ s'il existe une séquence de réduction ρ comportant k pas et ayant t pour source et t' pour cible. Remarquons que $k = 0$ correspond à $t = t'$.

Lemme 8.24 (Décomposition du *méta- λ -lifting*). Soient deux λ -termes étendus t et t' tels que $t \rightarrow_{lft} t'$. Alors il existe un entier $k \geq 0$ et un λ -terme étendu u tels que :

$$t \rightarrow_e^k u \rightarrow_c^{k+1} t'$$

Démonstration. Raisonnons par récurrence sur la longueur de la séquence formant la position étendue p du pas de réduction $t \xrightarrow{p}_{lft} t'$.

- Cas $t \xrightarrow{e}_{lft} t'$. Alors la réduction est par le cas *Réification* et $t \rightarrow_c t'$, ce qui valide la conclusion avec $k = 0$.
- Cas $t \xrightarrow{i:p}_{lft} t'$ avec $i \geq 1$. Alors la réduction est par le cas *Contexte* et la conclusion est immédiate par application de l'hypothèse de récurrence.

- Cas $t \xrightarrow{0:p}_{lft} t'$. Alors la réduction est par le cas *Méta* et nous avons plus précisément $t = \psi^{-1}(\lambda x.s)(t_1, \dots, t_n) \xrightarrow{0:p}_{lft} \psi^{-1}(\lambda x.s')(t_1, \dots, t_n) = t'$, avec la décomposition $s = \langle\langle s[t_1, \dots, t_n] \rangle\rangle^{(x)}$ et le pas de *méta- λ -lifting* $\lambda x.s[t_1, \dots, t_n] \xrightarrow{p}_{lft} \lambda x.s'[t_1, \dots, t_n]$. En particulier, pour tout i l'expression t_i est libre dans $\lambda x.s[t_1, \dots, t_n]$, donc $t \rightarrow_e \lambda x.s[t_1, \dots, t_n]$. De plus $s' = \langle\langle s'[t_1, \dots, t_n] \rangle\rangle^{(x)}$, donc $\lambda x.s'[t_1, \dots, t_n] \rightarrow_c t'$. Enfin, par hypothèse de récurrence il existe $k \geq 0$ tel que $\lambda x.s[t_1, \dots, t_n] \rightarrow_e^k \rightarrow_c^{k+1} \lambda x.s'[t_1, \dots, t_n]$, et nous pouvons conclure par composition de ces trois séquences de réduction. \square

La relation entre la contraction \rightarrow_c et l'expansion \rightarrow_e est également visible sur une inversion de leurs propriétés.

- Un radical de contraction ne peut jamais être créé par un pas de contraction, et tous les radicaux de contraction qu'un terme pourra connaître sont donc visibles dès l'origine. En revanche des radicaux peuvent être détruits, ce qui est en particulier la cause de la non-confluence de la contraction seule (vue aux exemples 8.14 et 8.15).
- En prenant le point de vue de l'expansion nous inversons création et destruction. En particulier aucun radical d'expansion ne peut jamais être détruit par un pas d'expansion, mais en revanche des radicaux peuvent être créés.

Enfin, nous pouvons montrer que le *méta- λ -lifting* en tant que système de réécriture à des propriétés fortes, et qu'il définit bien une transformation globale non-ambiguë.

Lemme 8.25 (Convergence). *La relation de méta- λ -lifting a la propriété du diamant et est fortement normalisante.*

Démonstration. Considérons deux pas de *méta- λ -lifting* $t \xrightarrow{p_1}_{lft} t_1$ et $t \xrightarrow{p_2}_{lft} t_2$ différents ayant pour source un même λ -terme étendu t . La clôture en diamant de ce diagramme est obtenue en raisonnant par cas sur les positions étendues p_1 et p_2 :

- Si p_1 et p_2 sont disjointes, alors les deux pas n'ont aucune interférence et nous avons $t_1 \xrightarrow{p_2}_{lft} t'$ et $t_2 \xrightarrow{p_1}_{lft} t'$.
- Si $p_1 < p_2$ et p_2 est une position étendue du squelette contracté par $t \xrightarrow{p_1}_{lft} t_1$, alors notons $p_2 = p_1 \cdot 1 \cdot p'_2$. Nous avons d'une part $t \xrightarrow{p_2}_{lft} t_2 \xrightarrow{p_1}_{lft} t'$ et d'autre part $t \xrightarrow{p_1}_{lft} t_1 \xrightarrow{p_1 \cdot 01 \cdot p'_2}_{lft} t'$, ou le dernier pas de *méta- λ -lifting* utilise une fois de plus la condition *Méta* que $t \xrightarrow{p_2}_{lft} t_2$.
- Si $p_1 < p_2$ et p_2 n'est pas une position étendue du squelette $\lambda x.s$ contracté par $t \xrightarrow{p_1}_{lft} t_1$, alors notons $p_2 = p_1 \cdot p_s \cdot p'_2$ où $s|_{p_s} = \square$. Alors nous avons $t_1 \xrightarrow{p_1 \cdot p'_2}_{lft} t'$ et $t_2 \xrightarrow{p_1}_{lft} t'$.

Prouvons maintenant la terminaison. Soit $\S_\lambda(\cdot)$ une mesure sur les λ -termes

étendus dénotant le nombre de radicaux \rightarrow_{lft} . Formellement :

$$\begin{aligned} \S_\lambda(x) &= 0 \\ \S_\lambda(\square) &= 0 \\ \S_\lambda(\lambda x.t) &= 1 + \S_\lambda(t) \\ \S_\lambda(t_1 t_2) &= \S_\lambda(t_1) + \S_\lambda(t_2) \\ \S_\lambda(f(t_1, \dots, t_n)) &= \S_\lambda(s) + \sum_i \S_\lambda(t_i) \quad \psi(f) = \lambda x.s \end{aligned}$$

Le point crucial de cette définition est l'omission dans le dernier cas de la λ -abstraction qui se trouve à la racine de $\psi(f)$.

Si $t \rightarrow_c t'$ alors $\S_\lambda(t') = \S_\lambda(t) - 1$, et si $t \rightarrow_e t'$ alors $\S_\lambda(t') = \S_\lambda(t) + 1$. Par décomposition de \rightarrow_{lft} sur \rightarrow_c et \rightarrow_e selon le lemme 8.24 nous déduisons donc que si $t \rightarrow_{lft} t'$ alors $\S_\lambda(t') = \S_\lambda(t) - 1$. En particulier la mesure décroît strictement et la relation de réduction \rightarrow_{lft} est fortement normalisante. \square

Avec ce résultat de convergence de la relation de réduction de *méta- λ -lifting* nous obtenons une fonction totale des λ -termes dans les termes du premier ordre sur la signature \mathcal{F} . Cette fonction totale représente la transformation globale de *λ -lifting* pleinement paresseux telle que décrite dans la littérature [Hug82, PJ87].

Nous pouvons également constater que, similairement au *méta- λ -lifting* mais contrairement à la contraction, l'expansion prise seule forme un système convergent (lemme 8.26). Par cela nous pouvons aussi définir pour tout λ -terme étendu une *forme totalement expansée* (définition 8.27) qui sera utile à la définition d'un étiquetage à la section 8.5.

Lemme 8.26 (Convergence). *La relation \rightarrow_e a la propriété du diamant et est fortement normalisante.*

Démonstration. La propriété du diamant est immédiate car deux pas d'expansion ne peuvent interférer d'aucune manière : chacun est attaché à un unique symbole et le remplace par un contexte clos, sans effacer ni dupliquer d'autres positions que la sienne.

Prouvons maintenant la terminaison, en introduisant une mesure $\S_f(\cdot)$ sur les λ -termes étendus majorant le nombre de radicaux d'expansion. Cette mesure compte le nombre de positions étendues dénotant un symbole de supercombinateur. Formellement :

$$\begin{aligned} \S_f(x) &= 0 \\ \S_f(\square) &= 0 \\ \S_f(\lambda x.t) &= \S_f(t) \\ \S_f(t_1 t_2) &= \S_f(t_1) + \S_f(t_2) \\ \S_f(f(t_1, \dots, t_n)) &= 1 + \S_f(\psi(f)) + \sum_i \S_f(t_i) \end{aligned}$$

Remarquons que cette mesure est bien définie car ψ est bien fondé par convention 8.10. De plus, pour tous deux λ -termes étendus t et u , si $t \rightarrow_e u$ alors $\S_f(u) = \S_f(t) - 1$. Donc la mesure décroît strictement et \rightarrow_e est bien fortement normalisante. \square

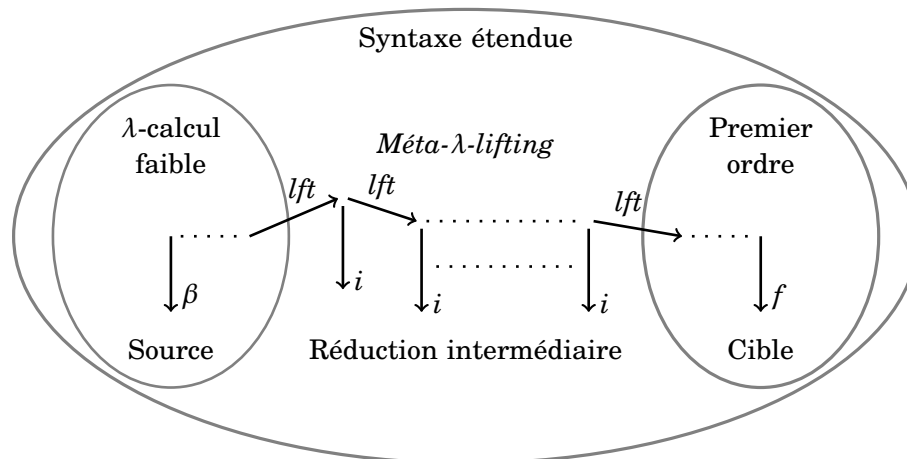
Définition 8.27. La *forme totalement expansée* d'un λ -terme étendu t est l'unique forme normale de t pour l'expansion.

Jusqu'ici nous n'avons vu que les aspects statiques du λ -lifting et des systèmes associés : le passage d'un terme à un autre terme. Nous allons voir dans la section suivante la partie dynamique de cette transformation, qui établit une bisimulation entre la réduction faible du λ -calcul et la réduction d'un système de réécriture du premier ordre orthogonal défini sur la signature \mathcal{F} .

8.3 Bisimulation

Cette section définit les règles d'évaluation des supercombinateurs en termes de règles de réécriture du premier ordre, et montre que le système du premier ordre orthogonal obtenu est bisimilaire au λ -calcul faible.

La relation de bisimulation témoignant de ce fait est la fonction globale de λ -lifting. Cependant, pour maintenir une preuve aussi simple que possible nous allons nous servir du fait que cette fonction globale est donnée par la relation de réduction d'un système de réécriture : le *méta- λ -lifting*. Et de la même manière que le *méta- λ -lifting* décrit une transformation progressive, « à petits pas », des termes, nous allons observer sous son action une transformation progressive, « à petits pas », des séquences de réduction. Ainsi pour obtenir la bisimulation il suffira de prouver que tout pas de *méta- λ -lifting* commute d'une certaine manière avec la β -réduction et les règles de notre système du premier ordre (lemme 8.32). Nous exprimerons ceci à l'aide d'une notion de *réduction intermédiaire* représentant à la fois la β -réduction faible *source* et la réduction du premier ordre *cible* sur les termes intermédiaires de la transformation.



La réduction cible a pour objectif de donner aux symboles de supercombinateurs le même comportement que les programmes qu'ils représentent. Elle consiste donc en un sens à récupérer et exécuter le code du sous-programme désigné par le symbole.

Définition 8.28 (Réductions source, cible, et intermédiaire). *Le schéma de réduction source est donné par la β -réduction usuelle appliquée aux λ -termes étendus :*

$$(\lambda x.t)u \rightarrow_{\beta} t^{x:=u}$$

Le schéma de réduction cible est :

$$\begin{aligned} @(f(t_1, \dots, t_n), u) \rightarrow_f s^{x:=u}[t_1, \dots, t_n] \quad \psi(f) = \lambda x.s \\ \forall i. t_i \text{ libre dans } \lambda x.s[t_1, \dots, t_n] \end{aligned}$$

Un pas de réduction source (resp. cible) est l'application du schéma de réduction source (resp. cible) à une position non gelée d'un λ -terme étendu. Nous appellerons réduction intermédiaire l'union $\rightarrow_i = \rightarrow_{\beta} \cup \rightarrow_f$ des réductions source et cible.

Remarquons que les réductions source, intermédiaire, et cible sont soumises à la restriction de réduction faible, contrairement au *méta- λ -lifting*.

La métaphore décrivant la réduction cible comme un appel à un sous-programme distant se traduit directement en la décomposition de la réduction cible en un pas d'expansion (« récupérer le code ») suivi d'un pas de β -réduction faible (« exécuter ce code »).

Lemme 8.29 (Décomposition de la réduction cible). *Soient deux λ -termes étendus t et t' tels que $t \rightarrow_f t'$. Alors il existe un λ -terme étendu v tel que :*

$$t \rightarrow_e v \rightarrow_{\beta} t'$$

Démonstration. Notons $t = c[@(f(t_1, \dots, t_n), u)] \rightarrow_f c[s^{x:=u}[t_1, \dots, t_n]] = t'$. Par définition $\psi(f) = \lambda x.s$, et pour tout i l'expression t_i est libre dans $\lambda x.s[t_1, \dots, t_n]$. En particulier l'expansion de f est définie et nous avons

$$t = c[@(f(t_1, \dots, t_n), u)] \rightarrow_e c[@(\lambda x.s[t_1, \dots, t_n], u)]$$

Notons $v = c[@(\lambda x.s[t_1, \dots, t_n], u)]$. La position de l'application n'étant pas gelée nous avons

$$v = c[@(\lambda x.s[t_1, \dots, t_n], u)] \rightarrow_{\beta} c[(s[t_1, \dots, t_n])^{x:=u}]$$

Or pour tout i l'expression t_i est libre dans $\lambda x.s[t_1, \dots, t_n]$, d'où en particulier $x \notin \text{fv}(t_i)$. Ainsi

$$c[(s[t_1, \dots, t_n])^{x:=u}] = c[s^{x:=u}[t_1, \dots, t_n]] = t'$$

□

Remarque 8.30. *La réduction cible peut être mise sous la forme d'un système de réécriture orthogonal, avec un ensemble de règles dont tous les membres gauches ont la forme $@(f(Z_1, \dots, Z_n), Z_0)$ avec un symbole $f \in \mathcal{F}$ différent pour chaque règle. Ceci sera formellement le cas au chapitre 9 suivant sur le lifting généralisé aux CRS faibles.*

Lemme 8.31 (Commutation avec la substitution).

- Si $t \rightarrow_{lft} t'$, alors $t^{x:=u} \rightarrow_{lft} t'^{x:=u}$.
- Si $u \rightarrow_{lft} u'$, alors $t^{x:=u} \rightarrow_{lft} t'^{x:=u}$.

Démonstration.

- Par induction sur la définition de $t \rightarrow_{lft} t'$, avec l'équivalent sur la syntaxe étendue du lemme 2.7 de stabilité des squelettes par substitution.
- Par induction sur la définition de $t^{x:=u}$. □

Lemme 8.32 (Commutation).

- Si
$$\begin{array}{ccc} t & \xrightarrow{lft} & u \\ i \downarrow & & \\ t' & & \end{array} \quad \text{alors il existe } u' \text{ tel que } \begin{array}{ccc} & & u \\ & & \downarrow i \\ t' & \xrightarrow{lft} & u' \end{array} .$$
- Si
$$\begin{array}{ccc} t & \xrightarrow{lft} & u \\ & & \downarrow i \\ & & u' \end{array} \quad \text{alors il existe } t' \text{ tel que } \begin{array}{ccc} & & t \\ i \downarrow & & \\ t' & \xrightarrow{lft} & u' \end{array} .$$

Démonstration. Supposons $t'_i \leftarrow t \rightarrow_{lft} u$, et raisonnons par cas sur les positions relatives des deux radicaux.

- I) Réductions à des positions disjointes. Ce cas est immédiat.
- II) Réduction \rightarrow_{lft} dans la fonction du radical \rightarrow_i . Raisonnons par cas sur \rightarrow_{lft} .
 - Réification $t = c[(\lambda x.s[t_1, \dots, t_n])a] \rightarrow_{lft} c[\psi^{-1}(\lambda x.s)(t_1, \dots, t_n)] = u$. Alors $u = c[\psi^{-1}(\lambda x.s)(t_1, \dots, t_n)] \rightarrow_f c[s^{x:=a}[t_1, \dots, t_n]] = t'$.
 - Méta $t = c[@(f(t_1, \dots, t_n), a)] \rightarrow_{lft} c[@(g(t_1, \dots, t_n), a)] = u$ avec le pas $\psi(f)[t_1, \dots, t_n] \rightarrow_{lft} \psi(g)[t_1, \dots, t_n]$. Notons $\psi(f) = \lambda x.s_f$ et $\psi(g) = \lambda x.s_g$. Nous avons d'une part $t = c[@(f(t_1, \dots, t_n), a)] \rightarrow_f c[s_f^{x:=a}[t_1, \dots, t_n]] = t'$ et d'autre part $t = c[@(g(t_1, \dots, t_n), a)] \rightarrow_f c[s_g^{x:=a}[t_1, \dots, t_n]]$. Notons $u' = c[s_g^{x:=a}[t_1, \dots, t_n]]$. De plus par définition $s_f[t_1, \dots, t_n] \rightarrow_{lft} s_g[t_1, \dots, t_n]$. Alors $s_f^{x:=a}[t_1, \dots, t_n] \rightarrow_{lft} s_g^{x:=a}[t_1, \dots, t_n]$ par lemme 8.31, et $t' \rightarrow_{lft} u'$.
 - Contexte. Il y a trois cas à considérer.
 - i) Réduction \rightarrow_{lft} dans le squelette de la fonction d'un radical \rightarrow_β . Alors $t = c[(\lambda x.s[t_1, \dots, t_n])a] \rightarrow_{lft} c[(\lambda x.s'[t_1, \dots, t_n])a] = u$ avec le pas $s[t_1, \dots, t_n] \rightarrow_{lft} s'[t_1, \dots, t_n]$, et $t' = c[s^{x:=a}[t_1, \dots, t_n]]$. Notons $u' = c[s'^{x:=a}[t_1, \dots, t_n]]$, nous avons de plus $u \rightarrow_\beta u'$. Par lemme 8.31 enfin $s^{x:=a}[t_1, \dots, t_n] \rightarrow_{lft} s'^{x:=a}[t_1, \dots, t_n]$, donc $t' \rightarrow_{lft} u'$.
 - ii) Réduction \rightarrow_{lft} dans une expression libre de la fonction d'un radical \rightarrow_β . Sans perte de généralité nous pouvons supposer être dans le cas $t = c[(\lambda x.s[t_1, \dots, t_n])a] \rightarrow_{lft} c[(\lambda x.s[t'_1, t_2, \dots, t_n])a] = u$, avec $t' = c[s^{x:=a}[t_1, \dots, t_n]]$. Alors notons $u' = c[s'^{x:=a}[t'_1, t_2, \dots, t_n]]$. Nous avons bien $u \rightarrow_\beta u'$. De plus $\langle\langle s[t'_1, t_2, \dots, t_n] \rangle\rangle^{x} = \langle\langle s[t_1, \dots, t_n] \rangle\rangle^{x} = s$ donc $t' \rightarrow_{lft} u'$.
 - iii) Réduction \rightarrow_{lft} dans un paramètre du supercombinateur d'un radical \rightarrow_f . Sans perte de généralité supposons $t = c[@(f(t_1, \dots, t_n), a)] \rightarrow_{lft}$

$c[@(f(t'_1, t_2, \dots, t_n), a)] = u$, avec $\psi(f) = \lambda x.s$ et $t' = c[s^{(x:=a)}[t_1, \dots, t_n]]$.
Alors notons $u' = c[s^{(x:=a)}[t'_1, t_2, \dots, t_n]]$. Nous avons bien $t' \rightarrow_{lft} u'_f \leftarrow u$.

III) Réduction \rightarrow_{lft} dans l'argument du radical \rightarrow_i .

- Cas d'un radical \rightarrow_β . Alors $t = c[(\lambda x.s[t_1, \dots, t_n])a] \rightarrow_{lft} c[(\lambda x.s[t_1, \dots, t_n])a'] = u$ et $t' = c[s^{(x:=a)}[t_1, \dots, t_n]]$. Notons $u' = t' = c[s^{(x:=a')}[t_1, \dots, t_n]]$. Nous avons bien $u \rightarrow_f u'$. De plus $t' \rightarrow_{lft} u'$.
- Cas d'un radical \rightarrow_f . Alors $t = c[@(f(t_1, \dots, t_n), a)] \rightarrow_{lft} c[@(f(t_1, \dots, t_n), a')] = u$ avec $\psi(f) = \lambda x.s$ et $t' = c[s^{(x:=a)}[t_1, \dots, t_n]]$. Notons $u' = c[s^{(x:=a')}[t_1, \dots, t_n]]$. Nous avons bien $u \rightarrow_f u'$. De plus $t' \rightarrow_{lft} u'$ par lemme 8.31.

IV) Réduction \rightarrow_i dans un paramètre du radical \rightarrow_{lft} .

- Réification. Alors $t = c[\lambda x.s[t_1, \dots, t_n]] \rightarrow_{lft} c[\psi^{-1}(\lambda x.s)(t_1, \dots, t_n)] = u$ et sans perte de généralité nous pouvons supposer $t_1 \rightarrow_i t'_1$ et $t' = c[\lambda x.s[t'_1, t_2, \dots, t_n]]$. Notons $u' = c[\psi^{-1}(\lambda x.s)(t'_1, t_2, \dots, t_n)]$. Nous avons $u \rightarrow_i u'$. De plus $\langle\langle s[t_1, \dots, t_n] \rangle\rangle^{(x)} = \langle\langle s[t'_1, t_2, \dots, t_n] \rangle\rangle^{(x)} = s$, donc $t' \rightarrow_{lft} u'$.
- Méta. Alors $t = c[f(t_1, \dots, t_n)] \rightarrow_{lft} c[g(t_1, \dots, t_n)] = u$ avec $\psi(f) = \lambda x.s_f$, $\psi(g) = \lambda x.s_g$, $s_f[t_1, \dots, t_n] \rightarrow_{lft} s_g[t_1, \dots, t_n]$. Sans perte de généralité supposons $t_1 \rightarrow_i t'_1$ et $t' = c[f(t'_1, t_2, \dots, t_n)]$. Notons $u' = c[g(t'_1, t_2, \dots, t_n)]$. Nous avons $u \rightarrow_i u'$. De plus $\langle\langle s_f[t_1, \dots, t_n] \rangle\rangle^{(x)} = \langle\langle s_f[t'_1, t_2, \dots, t_n] \rangle\rangle^{(x)} = s$, alors $s_f[t'_1, t_2, \dots, t_n] \rightarrow_{lft} s_g[t'_1, t_2, \dots, t_n]$, et $t' \rightarrow_{lft} u'$.

V) Réduction \rightarrow_i dans le squelette du radical \rightarrow_{lft} . Ce cas est interdit par la réduction faible, les positions d'un squelette étant gelées.

Le deuxième point (simulation arrière) est similaire. □

En composant les diagrammes de bisimulation locale donnés par le lemme 8.32 de commutation de la réduction intermédiaire avec le *méta- λ -lifting* nous obtenons des diagrammes de bisimulation sur les séquences de *méta- λ -lifting* :

$$\begin{array}{l}
 \text{– Si } \begin{array}{ccc} & \xrightarrow{lft} & u \\ i \downarrow & & \\ & \xrightarrow{lft} & t' \end{array} \text{ alors il existe } u' \text{ tel que } \begin{array}{ccc} & & u \\ & & \downarrow i \\ t' & \xrightarrow{lft} & u' \end{array} . \\
 \\
 \text{– Si } \begin{array}{ccc} t & \xrightarrow{lft} & u \\ & & \downarrow i \\ & & u' \end{array} \text{ alors il existe } t' \text{ tel que } \begin{array}{ccc} & & t \\ i \downarrow & & \\ t' & \xrightarrow{lft} & u' \end{array} .
 \end{array}$$

Autrement dit, un corollaire immédiat du lemme 8.32 est le théorème de bisimulation suivant.

Théorème 8.33. *La fonction de λ -lifting est une bisimulation entre la réduction source \rightarrow_β et la réduction cible \rightarrow_f .*

Ainsi, le λ -calcul muni de la β -réduction faible est bisimilaire à un système de réécriture du premier ordre orthogonal.

Enfin, il peut être prouvé que cette bisimulation préserve les notions de descendance, et donc également de résiduation et de nécessité. En outre, nous pouvons donner du λ -lifting une version étiquetée qui est un *SLS* et qui définit une bisimulation entre la réduction avec partage pleinement paresseux du λ -calcul faible et une certaine réduction partagée du système du premier ordre cible.

Nous ne détaillerons pas l'ensemble de ces points pour le λ -calcul, mais le chapitre 9 suivant donnera toutes les preuves dans le cadre plus général des *CRS* faibles. Nous nous contenterons à ce chapitre de montrer comment le *méta- λ -lifting* peut être étiqueté de manière à former un *SLS*. Ceci sera fait à la section 8.5, après un commentaire sur le lien entre réduction faible et orthogonalité.

8.4 Discussion : réduction faible et orthogonalité

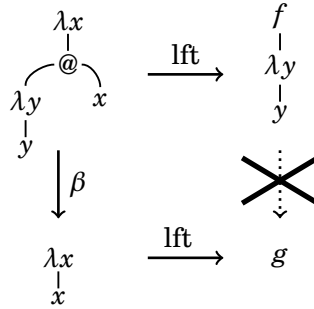
Remarquons que, comme l'illustre l'exemple 8.34 suivant, le théorème 8.33 de bisimulation est bien lié à la réduction faible et n'est plus valable en cas de réduction non-faible¹.

Exemple 8.34.

Considérons le λ -terme $\lambda x.(\lambda y.y)x$ et deux symboles f et g respectivement d'arité 1 et 0 et tels que

$$\begin{aligned}\psi(f) &= \lambda x.\Box x \\ \psi(g) &= \lambda x.x\end{aligned}$$

En considérant le *méta- λ -lifting* tel que défini à la section 8.2, la réduction cible (faible) telle que définie à la section 8.3, et la β -réduction non-faible nous obtenons les pas de réduction suivants :



Le terme $f(\lambda y.y)$ est une forme normale où aucun pas équivalent au pas de β -réduction non-faible $\lambda x.(\lambda y.y)x \rightarrow \lambda x.x$ n'existe.

Sur le modèle du *méta- λ -lifting* nous pourrions imaginer une réduction cible plus riche, qui puisse « défaire » certains pas *lifting* afin de révéler des radicaux cachés par les symboles de supercombinateurs. Dans le cadre de l'exemple 8.34 nous pourrions alors avoir un pas de réduction $f(\lambda y.y) \rightarrow g$.

1. L'effet de la restriction à la réduction faible apparaît également au cas V) de la preuve du lemme 8.32 de commutation entre réduction intermédiaire et *méta- λ -lifting*.

Cependant, l'exemple 8.35 suivant montre que nous y perdrons tout espoir d'orthogonalité pour le système cible.

Exemple 8.35.

Dans le cadre de l'exemple 8.34, considérons λ -terme étendu $@(f(\lambda y.y), a)$ où a est un symbole d'arité nulle, et observons les deux pas de réduction suivants, où sont soulignées les parties significatives des radicaux.

$$\begin{aligned} @(\underline{f(\lambda y.y)}, a) &\rightarrow @(g, a) \\ @(\underline{f(\lambda y.y)}, a) &\rightarrow @(\lambda y.y, a) \end{aligned}$$

Il y a un recouvrement au niveau du symbole f .

Ce phénomène met au jour une observation qui sera développée au chapitre 11 : vue à travers le filtre de la structuration des λ -termes en squelettes, la β -réduction non-faible n'apparaît pas orthogonale.

Or, nous réutiliserons le système cible au chapitre 10, et il convient donc de soigner sa définition. En particulier, la préservation de l'orthogonalité des systèmes qui sera établie au chapitre 9 suivant (théorème 9.37) sera un outil clé dans la mise en place du partage optimal. Nous ne pouvons donc pas nous permettre la rupture qu'impliquerait une réduction cible bisimilaire à la β -réduction non-faible.

8.5 *Lambda-lifting* étiqueté

Nous avons montré que le λ -*lifting* matérialisait un lien entre la β -réduction faible et un système de réécriture du premier ordre orthogonal. Nous allons maintenant voir que ce lien va au-delà de la simple réduction de termes, et couvre également la réduction partagée. Bien sûr nous utiliserons encore notre paradigme de représentation de la réduction partagée par des termes étiquetés, et ainsi la présente section adapte l'étiquetage du chapitre 2 à la syntaxe étendue donnée par la définition 8.6 et propose des versions étiquetées de la réduction source, de la réduction cible, et du *méta- λ -lifting*, de sorte que l'ensemble forme un *SLS*. Nous obtenons donc une notion de λ -*lifting* applicable aux λ -graphes de Wadsworth [Wad71]. Ainsi, la réduction de graphes liée au λ -*lifting* pleinement paresseux [PJ87] est assimilée à la notion unifiée de pleine paresse établie au chapitre 3.

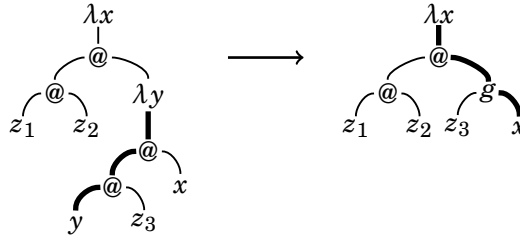
Le point décisif de la construction de cette section est l'invariant dit des *squelettes uniformes* (définition 8.44), qui une fois mis en place permettra de se reposer sur les idées déjà vues aux chapitres 2 et 4. Ce nouvel invariant a pour objectif de rendre l'étiquetage des squelettes robuste aux expansions et aux contractions qui peuvent les affecter.

Pour poser ce problème, rappelons-nous un point crucial des développements des chapitres précédents : les squelettes sont des éléments stables de la réduction faible (lemme 2.8 d'invariance des squelettes). La transposition de cette remarque au λ -calcul faible étiqueté faisait que, connaissant l'étiquetage initial,

nous pouvons à n'importe quel moment d'une séquence de réduction reconstruire toutes les étiquettes d'un squelette donné à partir d'une seule d'entre elles. C'est notamment ce qui a permis, via le lemme 2.24 des squelettes arborescents, de vérifier très simplement que le λ -calcul faible étiqueté du chapitre 2 définissait un *SLS*.

Or cette stabilité n'est plus valide dans le présent chapitre. Ou du moins, il faut en assouplir l'énoncé pour l'adapter à notre système plus riche. En effet, quand nous ajoutons au système des pas de *méta- λ -lifting* qui ne sont pas soumis à la restriction de réduction faible, les squelettes peuvent être modifiés. L'exemple 8.36 rappelle ceci.

Exemple 8.36.



Comme à l'exemple 8.14 rappelé par ce schéma, considérons le λ -terme (à gauche)

$$t = \lambda x.z_1 z_2 (\lambda y.y z_3 x)$$

et le symbole g d'arité 2 tel que

$$\psi(g) = \lambda y.y \square \square$$

Le squelette de la λ -abstraction racine du λ -terme t est (en gras dans le schéma)

$$s_t = \lambda x.\square (\lambda y.y \square x)$$

De plus, nous pouvons appliquer un pas de *méta- λ -lifting* au niveau de la λ -abstraction λy pour obtenir le λ -terme étendu (à droite dans le schéma)

$$u = \lambda x.z_1 z_2 g(z_3, x)$$

Le squelette de la λ -abstraction racine du λ -terme étendu u , qui descend du squelette s_t , est (en gras dans le schéma)

$$s_u = \lambda x.\square g(\square, x) \neq s_t$$

Les squelettes peuvent donc être modifiés par le *méta- λ -lifting*, et nous ne pourrions obtenir des preuves aussi simples qu'au chapitre 2 que si nous utilisons une méthode d'étiquetage qui soit compatible avec ces modifications. Nous tirerons pour cela parti des propriétés de convergence de l'expansion : toutes

les modifications possibles d'un squelette donné s par contraction ou expansion (et donc par *méta- λ -lifting*) sont des contractions d'une même forme totalement expansée (par lemme 8.26 de convergence). Nous allons donc définir un *étiquetage initial étendu* (définition 8.41) faisant référence à cette forme totalement expansée, et bâtir notre invariant des *squelettes uniformes* (définition 8.44) grâce à cette base. Rappelons d'abord la définition des étiquettes utilisées au chapitre 2 (définition 2.15).

Définition 8.37 (Étiquettes). *L'ensemble \mathcal{L} des étiquettes est défini par la grammaire suivante :*

$$\alpha ::= p \mid [\alpha, \alpha] \quad p \text{ position}$$

et muni de l'ordre \leq généré par la condition $\omega \leq [\omega, \alpha]$ pour toutes étiquettes $\alpha, \omega \in \mathcal{L}$.

Les λ -termes étendus étiquetés sont une extension immédiate des λ -termes étiquetés du chapitre 2 (définition 2.16).

Définition 8.38 (Lambda-termes étendus étiquetés). *La syntaxe des termes étiquetés est donnée par :*

$$t ::= x \mid \lambda^\alpha x.t \mid (tt)^\alpha \mid f^\alpha(t_1, \dots, t_n) \quad f \text{ d'arité } n$$

Les notions de contextes, positions, variables libres, substitutions, et squelettes sont encore directement adaptées. L'application étiquetée $(tt)^\alpha$ pourra encore être notée $@^\alpha(t, t)$, et comme à la définition 1.13 nous notons t^* le terme t privé de ses étiquettes.

Aux chapitres 2 et 5 l'étiquetage initial d'un λ -terme annotait chaque λ -abstraction et chaque application par sa position dans le λ -terme. Nous allons encore ici suivre cette idée pour l'étiquetage initial des λ -termes étendus, mais en faisant référence à la position qu'aurait chaque élément dans la forme totalement expansée (définition 8.27), que nous appellerons *position expansée*. En quelque sorte, nous tenons compte dans l'étiquetage initial des positions qui sont « cachées » par des symboles de supercombinateurs.

Définition 8.39 (Positions expansées). *Soient c un λ -contexte étendu et $p \in \text{pos}(c)$ une position de c . L'**expansion** de p relativement à c est la position $\psi_c(p)$ définie par :*

$$\begin{aligned} \psi_c(\epsilon) &= \epsilon \\ \psi_{\lambda x.c}(1 \cdot p) &= 1 \cdot \psi_c(p) \\ \psi_{c_1 c_2}(i \cdot p) &= i \cdot \psi_{c_i}(p) & i \in \{1, 2\} \\ \psi_{f(c_1, \dots, c_n)}(i \cdot p) &= \psi_{\psi(f)(q_i)} \cdot \psi_{c_i}(p) & i \in \{1, \dots, n\}, q_i \text{ position du } i^{\text{ème}} \text{ trou de } \psi(f) \end{aligned}$$

Remarquons que si la définition de l'expansion d'une position p relativement à un λ -contexte étendu c est guidée par la position p , c'est sur le λ -contexte étendu c qu'a lieu l'induction, et ce n'est que parce que ψ est bien fondé (convention 8.10) que cette expansion est bien définie. Enfin, nous pouvons noter que

cette définition tient compte même des positions contenues dans les éventuels symboles qui ne peuvent pas être expansés (voir remarque 8.21), mais ceci n'a pas d'impact.

Exemple 8.40.

Ainsi qu'à l'exemple 8.12, considérons le λ -terme étendu

$$t = f(z_1 z_2, z_3)$$

ainsi que les expansions suivantes où f et g sont d'arité 2 :

$$\begin{aligned}\psi(f) &= \lambda x. \square g(\square, x) \\ \psi(g) &= \lambda y. y \square \square\end{aligned}$$

Calculons l'expansion de la position 12 (celle de l'occurrence de variable z_2) relativement à t :

$$\begin{aligned}& \psi_{f(z_1 z_2, z_3)}(12) \\ &= \psi_{\lambda x. \square g(\square, x)}(11) \cdot \psi_{z_1 z_2}(2) \quad \text{car le premier trou du contexte } \lambda x. \square g(\square, x) \text{ a pour position 11} \\ &= 1 \cdot \psi_{\square g(\square, x)}(1) \cdot \psi_{z_1 z_2}(2) \\ &= 11 \cdot \psi_{\square}(\epsilon) \cdot \psi_{z_1 z_2}(2) \\ &= 11 \cdot \psi_{z_1 z_2}(2) \\ &= 112 \cdot \psi_{z_2}(\epsilon) \\ &= 112\end{aligned}$$

De même, $\psi_t(2) = 12112$ (où le calcul demande cette fois « d'ouvrir » un supercombinateur à l'intérieur d'un supercombinateur).

Avec cette notion d'expansion nous pouvons généraliser l'étiquetage initial du chapitre 2 (définition 2.17) aux λ -termes étendus.

Définition 8.41 (Étiquetage initial étendu). *Pour toute position p la fonction $i_p(\cdot)$ transforme un λ -terme étendu en un λ -terme étendu étiqueté, avec les règles suivantes :*

$$\begin{aligned}i_p(x) &= x \\ i_p(\lambda x. t) &= \lambda^p x. i_{p-1}(t) \\ i_p(t_1 t_2) &= (i_{p-1}(t_1) i_{p-2}(t_2))^p \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p-q_1}(t_1), \dots, i_{p-q_n}(t_n)) \quad \forall j. q_j = \psi_{f(t_1, \dots, t_n)}(j)\end{aligned}$$

Soit t un λ -terme. **L'étiquetage initial** de t est le λ -terme étiqueté $i_\epsilon(t)$, que nous notons encore $i(t)$.

Remarquons que dans le cas d'un λ -terme étendu ne comportant aucune occurrence de symbole de supercombinateur, la définition 8.41 précédente coïncide avec la définition 2.17. Ainsi l'étiquetage initial du présent chapitre est bien une généralisation de l'étiquetage initial du chapitre 2.

Exemple 8.42.

Ainsi qu'aux exemples 8.12 et 8.40, considérons le λ -terme étendu $t = f(z_1 z_2, z_3)$ et les expansions $\psi(f) = \lambda x. \square g(\square, x)$ et $\psi(g) = \lambda y. y \square \square$ des symboles f et g d'arité 2. Alors

$$i(t) = f^e((z_1 z_2)^{11}, z_3)$$

Le réétiquetage uniforme (définition 2.18) est lui immédiatement adapté à la syntaxe étendue.

Définition 8.43 (Réétiquetage uniforme). *Pour tout λ -contexte étendu étiqueté c et toute étiquette $\omega \in \mathcal{L}$, le **réétiquetage uniforme** $[\omega, c]$ de c par ω est défini par les équations suivantes.*

$$\begin{aligned} [\omega, \square] &= \square \\ [\omega, x] &= x \\ [\omega, \lambda^\alpha x. c] &= \lambda^{[\omega, \alpha]} x. [\omega, c] \\ [\omega, @^\alpha(c_1, c_2)] &= @^{[\omega, \alpha]}([\omega, c_1], [\omega, c_2]) \\ [\omega, f^\alpha(c_1, \dots, c_n)] &= f^{[\omega, \alpha]}([\omega, c_1], \dots, [\omega, c_n]) \end{aligned}$$

Nous pouvons énoncer l'invariant sur les λ -termes étendus étiquetés qui guidera la définitions des réductions étiquetées et permettra une vérification aisée des axiomes des *SLS*. Cet invariant des squelettes uniformes (définition 8.44 suivante) demande aux étiquettes de tout squelette de correspondre à un étiquetage initial étendu soumis à un certain nombre de réétiquetages uniformes.

Définition 8.44 (Propriété des squelettes uniformes). *Un λ -terme étendu étiqueté t a la **propriété des squelettes uniformes**, noté $\mathbb{U}(t)$, si tout squelette de t est de la forme $s = [\omega_n, \dots [\omega_1, i_p(s^*)]]$ où p est une position et s^* un squelette non étiqueté.*

Il ne reste plus qu'à définir les version étiquetées de nos réductions (source, cible, et *méta- λ -lifting*) de manière à ce que notre invariant soit respecté.

La réduction source étiquetée sera un transfert direct de la β -réduction étiquetée faible du chapitre 2 (définition 2.20). La définition de la réduction cible étiquetée en revanche appelle un élément nouveau : en expansant un symbole de supercombinateur la réduction cible introduit de nouvelles positions qui n'ont pas d'ancêtres visibles dont nous puissions récupérer les étiquettes. La réduction cible étiquetée à partir d'un radical $@^\omega(f^\alpha(t_1, \dots, t_n), u)$ va donc construire de nouvelles étiquettes pour les positions de $\psi(f)$.

Comme dans le cadre des *CRS* où ce cas de figure apparaissait déjà (définition 4.42), ces nouvelles étiquettes viendront d'une combinaison de l'étiquette racine ω et d'un étiquetage « initial » de $\psi(f)$. Nous avons quand même un élément en plus par rapport au chapitre 4 : l'étiquette α du symbole de supercombinateur f , donc nous allons nous servir comme graine pour donner à l'ensemble des éléments de $\psi(f)$ un étiquetage qui satisfasse la propriété

des squelettes uniformes. Pour cela nous décomposerons l'étiquette α sous une forme $[\omega_n, \dots [\omega_1, p]]$ et utiliserons un étiquetage initial étendu à partir de p , uniformément réétiqueté par les ω_j .

Définition 8.45 (Réductions étiquetées). *La réduction source étiquetée est définie par le schéma :*

$$@^\omega(\lambda^\alpha x.t, u) \rightarrow_{\beta_i} [\omega, s]^{x:=u}[t_1, \dots, t_n]$$

avec les conditions $t = s[t_1, \dots, t_n]$ et $s = \langle\langle t \rangle\rangle^{x}$. *La réduction cible étiquetée est définie par le schéma :*

$$@^\omega(f^{[\omega_n, \dots [\omega_1, p]]}(t_1, \dots, t_n), u) \rightarrow_{f_i} [\omega, [\omega_n, \dots [\omega_1, i_p(s)]]]^{x:=u}[t_1, \dots, t_n]$$

où $\psi(f) = \lambda x.s$. Encore une fois, ces réductions sont restreintes aux positions non gelées, et nous notons \rightarrow_i la réduction intermédiaire étiquetée résultant de l'union de \rightarrow_{β_i} et \rightarrow_{f_i} .

Exemple 8.46.

Considérons le λ -terme étendu étiqueté

$$t = @^\omega(f^{[\alpha, 11]}(@^\beta(z_1, z_2), z_3), a^\gamma)$$

avec comme à l'exemple 8.12

$$\psi(f) = \lambda x.@(\square, g(\square, x))$$

Notons $s = @(\square, g(\square, x))$. Alors

$$[\alpha, i_{11}(s)] = @^{[\alpha, 11]}(\square, g^{[\alpha, 112]}(\square, x))$$

et nous avons le pas de réduction cible étiquetée

$$t \rightarrow_{f_i} @^{[\omega, [\alpha, 11]]}(@^\beta(z_1, z_2), g^{[\omega, [\alpha, 112]]}(z_3, a^\gamma))$$

Grâce au lemme 8.24 de décomposition, la version étiquetée du *méta- λ -lifting* peut être définie par l'intermédiaire des version étiquetées de la contraction et de l'expansion. Ces contraction et expansion étiquetées utilisent comme base le λ -expanseur ψ et font en sorte, dans un sens comme dans l'autre, la même étiquette soit donnée à un symbole f et à la racine du squelette associé. Comme dans la réduction cible étiquetée le reste est obtenu uniformément à partir d'un étiquetage initial étendu de manière à assurer la propriété des squelettes uniformes.

Définition 8.47 (Contraction et expansion étiquetés). *La contraction étiquetée et l'expansion étiqueté sont deux schémas de réduction définis par :*

$$\begin{aligned} \lambda^\alpha x.s[t_1, \dots, t_n] &\rightarrow_c (\psi^{-1}(\lambda x.s^*))^\alpha(t_1, \dots, t_n) \\ &\quad s = \langle\langle s[t_1, \dots, t_n] \rangle\rangle^{x} \\ f^{[\omega_n, \dots [\omega_1, p]]}(t_1, \dots, t_n) &\rightarrow_e [\omega_n, \dots [\omega_1, i_p(\psi(f))]] [t_1, \dots, t_n] \\ &\quad \forall j. t_j^* \text{ libre dans } \psi(f)[t_1^*, \dots, t_n^*] \end{aligned}$$

Nous pouvons vérifier que les opérations de contraction et d'expansion commutent avec l'étiquetage initial étendu.

Lemme 8.48 (Étiquetage initial étendu et contraction). *Considérons t et u deux λ -termes étendus (non étiquetés) tels que $t \rightarrow_c u$ (pour la contraction non-étiquetée). Soit p une position. Alors $i_p(t) \rightarrow_c i_p(u)$ (pour la contraction étiquetée).*

Démonstration. Notons $t = c[\lambda x.s[t_1, \dots, t_n]] \xrightarrow{q}_c c[f(t_1, \dots, t_n)] = u$. Nous avons $i_p(u) = c_u[f^{p \cdot q}(i_{p \cdot q \cdot q_1}(t_1), \dots, i_{p \cdot q \cdot q_n}(t_n))]$ avec $q_i = \psi_{f(t_1, \dots, t_n)}(i)$, c'est-à-dire que q_i est l'expansion de la position du $i^{\text{ème}}$ trou de $\psi(f)$, qui est encore l'expansion de la position de t_i dans $\lambda x.s[t_1, \dots, t_n]$. Donc

$$i_p(t) = c_u[\lambda^{p \cdot q} x. i_{p \cdot q \cdot 1}(s)[i_{p \cdot q \cdot q_1}(t_1), \dots, i_{p \cdot q \cdot q_n}(t_n)]]$$

et finalement $i_p(t) \xrightarrow{q}_c i_p(u)$. □

Nous prouvons de même le résultat réciproque pour l'expansion.

Lemme 8.49 (Étiquetage initiaux et expansion). *Considérons t et u deux λ -termes étendus (non étiquetés) tels que $t \rightarrow_e u$ (pour l'expansion non-étiquetée). Soit p une position. Alors $i_p(t) \rightarrow_e i_p(u)$ (pour l'expansion étiquetée).*

Nous en déduisons également que la contraction étiquetée, l'expansion étiquetée, et donc aussi le *méta- λ -lifting* étiqueté, préservent la propriété des squelettes uniformes.

Lemme 8.50 (Propriété des squelettes uniformes). *Considérons un λ -terme étendu étiqueté t tel que $\mathbb{U}(t)$. Si u est un λ -terme étendu étiqueté qui vérifie $t \rightarrow_c u$, $t \rightarrow_e u$, ou $t \rightarrow_{\beta_l} u$, alors il vérifie aussi $\mathbb{U}(u)$.*

De plus, et comme dans le cadre non étiqueté (lemme 8.29), tout pas de réduction cible étiquetée peut être décomposé en un pas d'expansion étiqueté suivi d'un pas de réduction source étiqueté.

Remarque 8.51. *Soient t et t' deux λ -termes étendus étiquetés tels que $t \rightarrow_{f_l} t'$. Alors il existe un λ -terme étendu étiqueté u tel que*

$$t \rightarrow_e u \rightarrow_{\beta_l} t'$$

La vérification de la préservation de la propriété des squelettes uniformes par les réductions source, intermédiaire, et cible, peut donc être ramenée à la seule vérification pour la réduction source, qui est immédiate car la β -réduction faible étiquetée vérifie toujours l'invariance des squelettes telle qu'énoncée au lemme 2.23.

Lemme 8.52 (Propriété des squelettes uniformes). *Considérons un λ -terme étendu étiqueté t tel que $\mathbb{U}(t)$. Si t' est un λ -terme étendu étiqueté qui vérifie $t \rightarrow_{\beta_l} t'$ ou $t \rightarrow_{f_l} t'$, alors il vérifie aussi $\mathbb{U}(t')$.*

Cet invariant assuré pour toutes nos notions de réduction, nous pourrions prouver de même qu'à la section 8.3 que le λ -*lifting* étiqueté définit une bisimulation entre la réduction source étiquetée et la réduction cible étiquetée, pour peu que nous considérons des termes vérifiant la propriété des squelettes uniformes, à partir d'une vérification de commutation locale :

- Si $\cup(t)$ et $t'_i \leftarrow t \rightarrow_{lft} u$ alors il existe u' tel que $t' \rightarrow_{lft} u'_i \leftarrow u$.
- Si $\cup(t)$ et $t \rightarrow_{lft} u \rightarrow_i u'$ alors il existe t' tel que $t \rightarrow_i t' \rightarrow_{lft} u'$.

Ceci dit, l'intérêt des termes étiquetés réside ici en ce qu'ils représentent des graphes. Le vrai objectif de cette section porte donc sur ce qui représente la réduction de graphe, c'est-à-dire la réduction parallèle étiquetée. Vérifions donc que l'ensemble formé par la réduction source étiquetée, la réduction cible étiquetée, et le *méta- λ -lifting* étiqueté forme un *SLS*. Rappelons dans cette formalisation que les réductions source et cible sont soumises à la restriction de réduction faible, tandis que la réduction de *méta- λ -lifting* peut s'appliquer sans restriction.

Instanciation 8.53. *Considérons le triplet $(\mathcal{S}, \mathcal{T}, \mathcal{R})$ formé de :*

- la signature $\mathcal{S} = \mathcal{X} \cup \{\lambda x \mid x \in \mathcal{X}\} \cup \{\@\} \cup \mathcal{F}$,
- l'ensemble \mathcal{T} des termes étiquetés vérifiant la propriété des squelettes uniformes, et
- l'ensemble $\mathcal{R} = \mathcal{R}_\beta \cup \mathcal{R}_f \cup \mathcal{R}_{lft}$ où \mathcal{R}_β est l'ensemble des pas de réduction source étiquetée dont la source vérifie la propriété des squelettes uniformes, \mathcal{R}_f est l'ensemble des pas de réduction cible étiquetée dont la source vérifie la propriété des squelettes uniformes, et \mathcal{R}_{lft} est l'ensemble des pas de méta- λ -lifting dont la source vérifie la propriété des squelettes uniformes.

Pour tout pas de réduction $\rho \in \mathcal{R}_\beta$, notons

$$\rho : c[@^\omega(\lambda^\alpha x.s[t_1, \dots, t_n], u)] \rightarrow_{\beta_i} c[[\omega, s]^{(x:=u)}[t_1, \dots, t_n]]$$

et définissons :

- $\text{dex}(\rho) = @^\omega(\lambda^\alpha x.s[t_1, \dots, t_n], u)$
- $\text{duct}(\rho) = [\omega, s]^{(x:=u)}[t_1, \dots, t_n]$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \text{pos}(s) \setminus \text{fo}_x(s)$

Pour tout pas de réduction $\rho \in \mathcal{R}_f$, notons

$$\rho : c[@^\omega(f^{[\omega_n, \dots, [\omega_1, p]]}(t_1, \dots, t_n), u)] \rightarrow_{f_i} c[[\omega, [\omega_n, \dots, [\omega_1, i_p(s)]]]^{(x:=u)}[t_1, \dots, t_n]]$$

avec $\psi(f) = \lambda x.s$. et définissons :

- $\text{dex}(\rho) = @^\omega(f^{[\omega_n, \dots, [\omega_1, p]]}(t_1, \dots, t_n), u)$
- $\text{duct}(\rho) = [\omega, [\omega_n, \dots, [\omega_1, i_p(s)]]]^{(x:=u)}[t_1, \dots, t_n]$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \text{pos}(s) \setminus \text{fo}_x(s)$

Pour tout pas de réduction $\rho \in \mathcal{R}_{lft}$, notons

$$\rho : c[s[t_1, \dots, t_n]] \rightarrow_{lft} c[f^\alpha(t_1, \dots, t_n)]$$

avec s un certain contexte d'arité n (formé par un squelette ou un symbole de supercombinateur selon que la réduction est par Réification ou Méta) tel que $\tau(s) = \alpha$ et définissons :

- $\text{dex}(\rho) = s[t_1, \dots, t_n]$
- $\text{duct}(\rho) = f^\alpha(t_1, \dots, t_n)$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \{c\}$

Lemme 8.54. *Le système $\Sigma = (\mathcal{S}[\mathcal{L}], \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.*

Démonstration.

Source & Cible Par lemmes 8.50 et 8.52 de préservation de la propriété des squelettes uniformes.

Résidus Par stabilité des positions non gelées, lemme 2.10.

Zone d'effet De même qu'à la section 2.3. □

Lemme 8.55. *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS*

Démonstration.

Étiquette racine Les applications, les λ -abstractions, et les symboles de supercombinateurs sont tous étiquetés.

Progression Dans le cas d'un pas de réduction ρ de \mathcal{R}_β ou \mathcal{R}_f , toute étiquette de la zone d'effet est de la forme $[\omega, \alpha]$ avec $\omega = \tau(\rho)$ et en particulier $\omega < [\omega, \alpha]$. Dans le cas d'un pas de réduction $\rho \in \mathcal{R}_{ift}$ la zone d'effet est un singleton dont l'unique étiquette est $\tau(\rho)$.

Héritage De même qu'à la section 2.3.

Partage Par propriété des squelettes uniformes, les étiquettes de deux positions différentes de la zone d'effet ne peuvent être égales dans le cas d'une réduction ρ de \mathcal{R}_β ou \mathcal{R}_f . Dans le cas d'une réduction $\rho \in \mathcal{R}_{ift}$ la zone d'effet est un singleton. □

Par cette représentation de la transformation de λ -lifting dans un SLS nous obtenons une notion de λ -lifting sur des graphes. Il est également possible, via ce λ -lifting étiqueté, de vérifier que ce lien entre la réduction partagée pleinement paresseuse connue sur le λ -calcul faible (chapitre 2) et la réduction partagée du système du premier ordre cible est encore une bisimulation, c'est-à-dire que nous avons le « même » partage avant et après λ -lifting (d'où l'équivalence avec les systèmes pleinement paresseux du chapitre 3). Formaliser vraiment ce nouveau résultat de bisimulation demande cependant d'établir un lien un peu plus précis entre les positions des termes avant et après λ -lifting.

L'opération est assez simple : il suffirait de définir une relation de descendance le long du *méta- λ -lifting* ou même simplement de la contraction et de vérifier que cette nouvelle notion se comporte bien et est compatible avec les étiquettes. Toutefois, cela serait encore du travail supplémentaire, puisque la

contraction, l'expansion, et le *méta- λ -lifting* sont de nouvelles structures que nous avons ajoutées au noyau du λ -calcul. En revanche, au chapitre 9 suivant, dans lequel nous allons généraliser la notion de *λ -lifting* aux *CRS*, nous pourrons exprimer toutes nos transformations comme des règles ordinaires du système, pour lesquelles il ne sera plus nécessaire de faire toutes ces définitions additionnelles. Nous pourrons en particulier formaliser les propriétés qui n'ont été qu'esquissées dans le présent chapitre.

Bilan du chapitre

Dans ce chapitre nous avons proposé une formalisation de la technique de compilation classique de *λ -lifting* pleinement paresseux en tant que transformation du λ -calcul vers un système de réécriture du premier ordre orthogonal, et prouvé que cette transformation définissait une bisimulation entre le λ -calcul faible et le système du premier ordre cible (théorème 8.33 de bisimulation).

Pour faciliter cette preuve nous avons décrit l'opération de *λ -lifting* comme une transformation « à petits pas » qui est un système de réécriture convergent dans une syntaxe étendue mélangeant les constructions du λ -calcul avec des symboles du premier ordre (section 8.2).

Enfin, nous avons également décrit une deuxième version, étiquetée, du *λ -lifting*, qui permet de compléter le chapitre 3 en faisant le lien entre la réduction pleinement paresseuse du λ -calcul du chapitre 2 et un certain partage du système du premier ordre cible.

Cette vision du *λ -lifting* comme un système de réécriture opérant une transformation entre deux systèmes de réécriture d'ordres différents va être reproduite et généralisée au chapitre 9 suivant pour lier l'ensemble des *CRS* faibles à des systèmes de réécriture du premier ordre, d'une manière qui préserve le partage, l'éventuelle orthogonalité des systèmes, et les notions de descendance et de résiduation. Ceci permettra au chapitre 10 de se ramener au premier ordre pour établir des résultats d'optimalité sur le partage pleinement paresseux.

CHAPITRE 9

Lifting généralisé

Sommaire

9.1 <i>Lifting</i> généralisé	280
9.2 Réductions source, intermédiaire, et cible	286
9.3 Bisimulation	291
9.4 Application à un système étiqueté	295

La transformation de λ -*lifting* a été introduite à l'origine comme un outil d'implémentation pour les langages de programmation fonctionnelle [Joh85, PJ87]. Nous l'avons également vue au chapitre 8 précédent comme la transformation d'un système de réécriture d'ordre supérieur orthogonal (le λ -calcul faible) en un système de réécriture du premier ordre orthogonal qui lui est bisimilaire.

Dans le présent chapitre nous allons généraliser la transformation aux *CRS* faibles, et montrer que le *lifting* des abstractions définit une bisimulation entre la réduction faible d'un *CRS* et un système de réécriture du premier ordre. Cette bisimulation préservera en outre le partage pleinement paresseux, et les notions de descendance, de résiduation et de nécessité. Enfin, la transformation préservera l'éventuelle orthogonalité du système source : en dépit du fait que nous utilisons comme étape intermédiaire un raffinement canonique qui casse l'orthogonalité stricte pour la remplacer par une version amoindrie (ladite « orthogonalité en réduction faible », définition 4.33), tout *CRS* faible orthogonal sera associé par *lifting* à un système du premier ordre orthogonal « tout court ».

La généralisation du *lifting* au cadre des *CRS* sera, comme on peut s'y attendre, source de difficultés nouvelles. Les règles de réécriture des *CRS* notamment sont bien plus riches que la seule règle β , et nous devons donc complexifier en particulier les notions de réduction cible et de réduction intermédiaire pour en tenir compte.

Cependant, la généralisation au cadre des *CRS* sera également une source de simplifications importantes des formalismes du chapitre 8. En effet nous allons pouvoir jouer jusqu'au bout le jeu de l'uniformisation du formalisme dans le cadre général des *CRS*. Alors que dans le chapitre 8 précédent nous avons défini deux notions distinctes de λ -*lifting*, avec ou sans étiquettes, dans

le présent chapitre une seule procédure de *lifting* sera définie pour tous les *CRS*. Cette procédure unique s'appliquera aussi bien à un *CRS* « brut » qu'à sa version étiquetée, en particulier grâce à l'internalisation des mécanismes d'étiquetage que nous avons mis en place pour les *CRS* dès le chapitre 4. Enfin, le formalisme unique des *CRS* couvrira à la fois le système source, le système cible, et la réduction intermédiaire, avec ou sans étiquettes, et même l'opération de *lifting*, tout cela sans qu'il soit nécessaire de redéfinir syntaxe, descendance, et autres notions utiles. En particulier, les définitions répétitives qui avaient pour certaines été omises au chapitre 8 seront ici immédiatement acquises, et nous pourrons compléter les énoncés et démonstrations qui y avaient été laissés à l'état d'allégations informelles.

La section 9.1 définit une transformation de *lifting* généralisant le λ -*lifting* aux termes d'un *CRS*, puis la section 9.2 donne la partie dynamique de cette transformation en définissant des notions de réduction intermédiaire et de réduction cible. La section 9.3 apporte la preuve que la transformation définit une bisimulation entre tout *CRS* faible source et le système du premier ordre cible associé, et démontre la préservation de l'éventuelle orthogonalité du système. La section 9.4 enfin montre comment le formalisme général de *lifting* s'accommode de l'étiquetage causal du chapitre 7, et préserve le partage ainsi que la descendance et la résiduation.

9.1 *Lifting* généralisé

Comme au chapitre 8 précédent nous allons baser le *lifting* sur un ensemble dénombrable de symboles de toutes arités s'ajoutant au système source et représentant les supercombinateurs. Et de même encore, une fonction d'expansion donnera une correspondance entre ces nouveaux symboles et des fragments de termes d'ordre supérieur ou du premier ordre.

Dès cet endroit, nous allons tirer parti du formalisme des *CRS* pour obtenir une formalisation moins *ad hoc* du *lifting*. Au chapitre 8, les nouveaux symboles demandaient la définition d'une nouvelle syntaxe, ce qui n'était pas très difficile mais demandait des adaptations en chaîne de toutes les définitions subséquentes. Ici, l'ensemble de symboles de supercombinateurs viendra simplement s'ajouter à la signature.

Également, pour obtenir une formalisation un peu plus abstraite et modulaire, dans ce chapitre l'ensemble de symboles \mathcal{F} comme la fonction d'expansion ψ seront des paramètres de l'opération de *lifting*. Alors, de même que pour un résultat axiomatique, une condition sera associée aux paramètres \mathcal{F} et ψ pour assurer la bonne formation du *lifting* (stabilité, définition 9.9). Ainsi par exemple, l'ajout d'étiquettes à une opération de *lifting* (section 9.4) ne sera pas un nouveau formalisme mais un nouveau choix de paramètres \mathcal{F} et ψ , et l'ensemble des constructions et des résultats sera alors déduit de la seule vérification de la conditions de stabilité (lemme 9.42).

Définition 9.1 (Ensemble de supercombinateurs). *Un ensemble de supercombinateurs \mathcal{F} est la donnée pour tout entier $n \in \mathbb{N}$ d'un ensemble dénombrable*

\mathcal{F}_n de symboles d'arité n telle que $\mathcal{F} = \bigcup_n \mathcal{F}_n$.

Pour toute signature de CRS \mathcal{S} et tout ensemble de supercombinateurs \mathcal{F} nous notons $\mathcal{S} \sqcup \mathcal{F}$ l'union disjointe de ces deux ensembles de symboles. L'ensemble obtenu est en particulier une signature de CRS. Les notions de termes, contextes, positions, variables libres, substitutions, et squelettes n'ont donc pas besoin d'être redéfinies lorsque nous étendons une signature \mathcal{S} en $\mathcal{S} \sqcup \mathcal{F}$.

Nous prenons comme au chapitre 8 la notation suivante.

Définition 9.2 (Ensemble des squelettes). *Pour toute signature \mathcal{S} , notons $\langle\langle \mathcal{S} \rangle\rangle$ l'ensemble des squelettes sur \mathcal{S} :*

$$\langle\langle \mathcal{S} \rangle\rangle = \{ \langle x \rangle \langle t \rangle^x \mid x \in \mathcal{X}, t \text{ un terme sur } \mathcal{S} \}$$

Avec des étiquettes (section 9.4) nous noterons encore

$$\langle\langle \mathcal{S}^{\mathcal{L}} \rangle\rangle = \{ \langle x \rangle^\alpha \langle t \rangle^x \mid x \in \mathcal{X}, \alpha \in \mathcal{L}, t \text{ un terme sur } \mathcal{S}^{\mathcal{L}} \}$$

Définition 9.3 (Expanseur). *Soient \mathcal{S} une signature CRS et \mathcal{F} un ensemble de supercombinateurs. Un **expanseur** de \mathcal{F} sur \mathcal{S} est une fonction injective $\psi : \mathcal{F} \rightarrow \langle\langle \mathcal{S} \sqcup \mathcal{F} \rangle\rangle$. Pour tout expanseur ψ de \mathcal{F} sur \mathcal{S} et tous symboles $f, g \in \mathcal{F}$ nous disons que g **contient** f par ψ et nous notons $g \supset_\psi f$ quand f apparaît dans $\psi(g)$. Un expanseur ψ est **bien fondé** quand il n'existe pas de suite infinie décroissante $f_1 \supset_\psi f_2 \supset_\psi \dots$*

Définition 9.4 (Système de lifting). *Soit \mathcal{S} une signature CRS. Un **système de lifting** pour \mathcal{S} est une paire (\mathcal{F}, ψ) où*

- \mathcal{F} est un ensemble de supercombinateurs.
- ψ est un expanseur bien fondé de \mathcal{F} sur \mathcal{S} .

Étant donné un système de lifting (\mathcal{F}, ψ) pour une signature \mathcal{S} nous pouvons définir comme au chapitre précédent une notion de *position étendue* sur la signature étendue $\mathcal{S} \sqcup \mathcal{F}$.

Définition 9.5 (Positions étendues). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . L'ensemble des **positions étendues** par ψ d'un méta-terme t sur $\mathcal{S} \sqcup \mathcal{F}$, ou **ψ -positions** de t , noté $\text{pos}_\psi(t)$, est défini par :*

$$\begin{aligned} \text{pos}_\psi(x) &= \{\epsilon\} \\ \text{pos}_\psi(\langle x \rangle t) &= \{\epsilon\} \cup 1 \cdot \text{pos}_\psi(t) \\ \text{pos}_\psi(f(t_1, \dots, t_n)) &= \{\epsilon\} \cup (\bigcup_i i \cdot \text{pos}_\psi(t_i)) & f \in \mathcal{S} \\ \text{pos}_\psi(Z(t_1, \dots, t_n)) &= \{\epsilon\} \cup (\bigcup_i i \cdot \text{pos}_\psi(t_i)) \\ \text{pos}_\psi(f(t_1, \dots, t_n)) &= \{\epsilon\} \cup (0 \cdot \text{pos}_\psi(\psi(f))) \cup (\bigcup_i i \cdot \text{pos}_\psi(t_i)) & f \in \mathcal{F} \end{aligned}$$

Remarquons que cet ensemble est fini car ψ est bien fondé.

Comme au chapitre précédent, le *lifting* est basé sur deux opérations réciproques de contraction et d'expansion. Nous allons cependant y voir une

différence : au chapitre 8 les opérations de *lifting* étaient des schémas de réduction *ad hoc* définis sur les λ -termes étendus, avec des conditions de bord sur les variables de certains sous-termes. Ici, la contraction et l'expansion seront définies par des règles *CRS* ordinaires, et les conditions de sûreté des valuations contiendront toutes les conditions de bord utiles.

Définition 9.6 (Contraction/Expansion). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . La ψ -contraction et la ψ -expansion sont deux ensembles de règles *CRS* sur $\mathcal{S} \sqcup \mathcal{F}$ définis par :*

$$\begin{array}{lll} (\psi\text{-contraction}) & \langle x \rangle s[Z_1, \dots, Z_n] \rightarrow_c \psi^{-1}(\langle x \rangle s)(Z_1, \dots, Z_n) & \langle x \rangle s \in \langle\langle \mathcal{S} \sqcup \mathcal{F} \rangle\rangle \\ (\psi\text{-expansion}) & f(Z_1, \dots, Z_n) \rightarrow_e \psi(f)[Z_1, \dots, Z_n] & f \in \mathcal{F} \end{array}$$

Un **pas de contraction/d'expansion** est l'application d'une règle de contraction / d'expansion à n'importe quelle position dans un terme, un contexte, ou un méta-terme dont les méta-variables sont toutes d'arité nulle.

Lemme 9.7 (Contraction/Expansion). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . Alors pour tous termes t et u sur $\mathcal{S} \sqcup \mathcal{F}$ nous avons $t \rightarrow_c u$ si et seulement si $u \rightarrow_e t$.*

Démonstration. Considérons un pas de réduction $t = c[l^\sigma] \rightarrow_c c[r^\sigma] = u$ par une règle $l = \langle x \rangle s[Z_1, \dots, Z_n] \rightarrow_c f(Z_1, \dots, Z_n) = r$. Par définition d'un expanseur, $f = \psi^{-1}(\langle x \rangle s) \in \mathcal{F}$, donc $r = f(Z_1, \dots, Z_n) \rightarrow_e \langle x \rangle s(Z_1, \dots, Z_n) = l$ est une règle d'expansion. En outre, σ est une valuation sûre pour $l \rightarrow r$, donc σ est sûre pour l et pour r , donc σ est sûre pour $r \rightarrow l$, et ainsi $u = c[r^\sigma] \rightarrow_e c[l^\sigma] = t$.

De même, si $u \rightarrow_e t$ alors $t \rightarrow_c u$. □

L'expanseur d'un système de *lifting* n'est pas surjectif en général. La contraction peut donc ne pas s'appliquer à n'importe quelle abstraction. Dans la section 8.5 ceci n'avait pas d'incidence car la contraction étiquetée était définie sur un échantillon suffisamment dense de squelettes étiquetés (en particulier, son domaine de définition contenait au moins un étiquetage de chaque squelette non-étiqueté). Surtout, l'échantillon en question était stable par toutes les réductions.

Pour toute la suite nous utiliserons donc des *systèmes de lifting stables*, dans lesquels nous tiendrons pour axiome que le codomaine de l'expanseur est un ensemble suffisamment grand et suffisamment stable de squelettes. Nous allons pour cela définir un invariant sur les termes exprimant que tout squelette peut être contracté, et demander à cet invariant d'être préservé à la fois par la contraction, l'expansion, et les réductions du *CRS* source. La notion de stabilité pour un système de *lifting* ne dépend donc pas que de la signature à laquelle s'applique le système de *lifting*, mais de tout le *CRS* avec lequel le système est utilisé.

Définition 9.8 (Propriété de contractabilité). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . Un terme t sur $\mathcal{S} \sqcup \mathcal{F}$ à la **propriété de contractabilité** relativement à ψ , noté $C_\psi(t)$, si tout squelette de t est dans le codomaine de ψ .*

Définition 9.9 (Système de lifting stable). Soit $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS faible. Un **système de lifting stable** pour Σ est un système de lifting (\mathcal{F}, ψ) pour \mathcal{S} tel que la propriété \mathbb{C}_ψ de contractabilité relativement à ψ est préservée par :

- la réduction faible générée par \mathcal{R} ,
- la ψ -contraction, et
- la ψ -expansion.

Remarquons que, comme au chapitre 8, la réduction source est soumise à la restriction de réduction faible mais pas les réductions de contraction ou d'expansion.

Nous pouvons définir une relation de réécriture convergente effectuant le *lifting*, de manière similaire à ce qui a été fait pour le *méta- λ -lifting* pour le λ -calcul au chapitre 8 (définition 8.17). Cette relation se décomposera toujours sur la contraction et l'expansion. En revanche, le *méta-lifting* que nous allons voir maintenant va être défini par des règles CRS ordinaires. Ces règles seront définies par *rangs*, le rang 0 correspondant au cas *Réification* et les rangs à partir de 1 au cas *Méta*.

Définition 9.10 (*Méta-lifting*). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ .

- Les **règles de méta-lifting de rang 0** sont les règles de contraction.
- Une paire $f(Z_1, \dots, Z_n) \rightarrow f'(Z_1, \dots, Z_n)$ est une **règle de méta-lifting de rang $n + 1$** si et seulement si $\psi(f) \rightarrow \psi(f')$ par une règle de méta-lifting de rang n .

Comme pour la contraction et l'expansion, ces règles peuvent s'appliquer à n'importe quelle position dans un terme, un contexte, ou un méta-terme dont les méta-variables sont toutes d'arité nulle. L'ensemble des **règles de méta-lifting** est l'union des ensembles des règles de méta-lifting de tous rangs $n \in \mathbb{N}$.

Comme à la définition 8.17 nous pouvons associer des positions étendues aux pas de *méta-lifting*.

Définition 9.11 (Position d'un pas de *méta-lifting*). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ .

Réification Si $t \rightarrow_{lft} t'$ par l'application à la position p d'une règle de rang 0, alors nous notons $t \xrightarrow{p}_{lft} t'$.

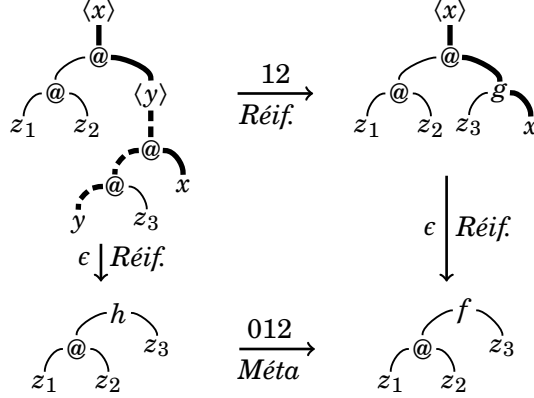
Méta Si $t \rightarrow_{lft} t'$ par l'application à la position p d'une règle $f(Z_1, \dots, Z_n) \rightarrow f'(Z_1, \dots, Z_n)$ de rang strictement supérieur à 0 et si q est la position étendue telle que $\psi(f) \xrightarrow{q}_{lft} \psi(f')$, alors nous notons $t \xrightarrow{p \cdot 0 \cdot q}_{lft} t'$.

Exemple 9.12.

Soit $t = \langle x \rangle @ (@(z_1, z_2), \langle y \rangle @ (@(y, z_3), x))$ un terme. Notons :

$$\begin{aligned} \varphi^{-1}(h) &= \langle x \rangle @ (\square, \langle y \rangle @ (@(y, \square), x)) \\ \varphi^{-1}(g) &= \langle y \rangle @ (@(y, \square), \square) \\ \varphi^{-1}(f) &= \langle x \rangle @ (\square, g(\square, x)) \end{aligned}$$

Les lignes pointillées désignent ci-dessous le squelette de l'abstraction $\langle y \rangle$, et les lignes grasses (pleines ou pointillées) désignent le squelette de l'abstraction $\langle x \rangle$.



Le lemme 9.13 suivant exprime la décomposition du *méta-lifting* sur les opérations de contraction et d'expansion. Il est la traduction directe du lemme 8.24 sur le *méta- λ -lifting* et se prouve de même.

Lemme 9.13 (Décomposition du *méta-lifting*). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Soient deux termes sur $\mathcal{S} \sqcup \mathcal{F}$ t et t' tels que $t \rightarrow_{\text{ift}} t'$. Alors il existe un entier $k \geq 0$ et un terme sur $\mathcal{S} \sqcup \mathcal{F}$ u tels que :

$$t \xrightarrow{e}^k u \xrightarrow{c}^{k+1} t'$$

En faisant appel aux même argument que lors de la preuve du lemme 8.25, nous pouvons montrer que le *méta-lifting* est un CRS convergent.

Lemme 9.14 (Convergence). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . La relation de méta-lifting a la propriété du diamant et est fortement normalisante.

Démonstration. Considérons deux pas de *méta-lifting* $t \xrightarrow{p_1}_{\text{ift}} t_1$ et $t \xrightarrow{p_2}_{\text{ift}} t_2$ différents ayant pour source un même terme t . La clôture en diamant de ce diagramme est obtenue en raisonnant par cas sur les positions étendues p_1 et p_2 :

- Si p_1 et p_2 sont disjointes, alors les deux pas n'ont aucune interférence et nous avons $t_1 \xrightarrow{p_2}_{\text{ift}} t'$ et $t_2 \xrightarrow{p_1}_{\text{ift}} t'$.
- Si $p_1 < p_2$ et p_2 est une position étendue du squelette contracté par $t \xrightarrow{p_1}_{\text{ift}} t_1$, alors notons $p_2 = p_1 \cdot 1 \cdot p'_2$. Nous avons d'une part $t \xrightarrow{p_2}_{\text{ift}} t_2 \xrightarrow{p_1}_{\text{ift}} t'$ et d'autre part $t \xrightarrow{p_1}_{\text{ift}} t_2 \xrightarrow{p_1 \cdot 01 \cdot p'_2}_{\text{ift}} t'$, ou le dernier pas de *méta-lifting* utilise une fois de plus la condition *Méta* que $t \xrightarrow{p_2}_{\text{ift}} t_2$.
- Si $p_1 < p_2$ et p_2 n'est pas une position étendue du squelette $\langle x \rangle$ s contracté par $t \xrightarrow{p_1}_{\text{ift}} t_1$, alors notons $p_2 = p_1 \cdot p_s \cdot p'_2$ où $s|_{p_s} = \square$. Alors nous avons $t_1 \xrightarrow{p_1 \cdot p'_2}_{\text{ift}} t'$ et $t_2 \xrightarrow{p_1}_{\text{ift}} t'$.

Prouvons maintenant la terminaison. Soit $\S_\lambda(\cdot)$ une mesure sur les termes dénotant le nombre de radicaux \rightarrow_{lft} . Rappelons que sont également autorisées les méta-variables d'arité 0. Formellement :

$$\begin{aligned}\S_\lambda(x) &= 0 \\ \S_\lambda(\square) &= 0 \\ \S_\lambda(Z) &= 0 \\ \S_\lambda(\langle x \rangle t) &= 1 + \S_\lambda(t) \\ \S_\lambda(f(t_1, \dots, t_n)) &= \sum_i \S_\lambda(t_i) && f \in \mathcal{S} \\ \S_\lambda(f(t_1, \dots, t_n)) &= \S_\lambda(s) + \sum_i \S_\lambda(t_i) && f \in \mathcal{F} \text{ et } \psi(f) = \langle x \rangle s\end{aligned}$$

Le point crucial de cette définition est l'omission dans le dernier cas de l'abstraction qui se trouve à la racine de $\psi(f)$.

Si $t \rightarrow_c t'$ alors $\S_\lambda(t') = \S_\lambda(t) - 1$, et si $t \rightarrow_e t'$ alors $\S_\lambda(t') = \S_\lambda(t) + 1$. Par décomposition de \rightarrow_{lft} sur \rightarrow_c et \rightarrow_e nous déduisons donc que si $t \rightarrow_{lft} t'$ alors $\S_\lambda(t') = \S_\lambda(t) - 1$. Donc la mesure décroît strictement et la relation de réduction \rightarrow_{lft} est fortement normalisante. \square

Les règles de *méta-lifting* définissent donc une fonction totale des termes CRS sur un signature \mathcal{S} dans les termes du premier ordre sur une signature $\mathcal{S} \sqcup \mathcal{F}$, et cette transformation généralise celle déjà connue pour le λ -calcul.

Convention 9.15. Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Pour tout terme t sur la signature étendue $\mathcal{S} \sqcup \mathcal{F}$ nous notons $lft(t)$ la forme normale de t par méta-lifting.

Enfin, comme au chapitre 8 sur le λ -calcul, nous pouvons aussi vérifier que l'expansion est aussi convergente et définit une deuxième fonction inversant le lifting, définie des termes CRS sur une signature étendue $\mathcal{S} \sqcup \mathcal{F}$ vers les termes CRS sur une signature \mathcal{S} .

Lemme 9.16 (Convergence). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . La relation \rightarrow_e a la propriété du diamant et est fortement normalisante.

Démonstration. La propriété du diamant est immédiate car les règles d'expansion forment un système orthogonal.

Prouvons maintenant la terminaison, en introduisant une mesure $\S_f(\cdot)$ sur les λ -termes étendus majorant le nombre de radicaux d'expansion. Cette mesure compte le nombre de positions étendues dénotant un symbole de supercombinateur. Formellement :

$$\begin{aligned}\S_f(x) &= 0 \\ \S_f(\square) &= 0 \\ \S_f(Z) &= 0 \\ \S_f(\langle x \rangle t) &= \S_f(t) \\ \S_f(f(t_1, \dots, t_n)) &= \S_f(\psi(f)) + \sum_i \S_f(t_i) && f \in \mathcal{S} \\ \S_f(f(t_1, \dots, t_n)) &= 1 + \S_f(\psi(f)) + \sum_i \S_f(t_i) && f \in \mathcal{F}\end{aligned}$$

Pour tous termes t et u , si $t \rightarrow_e u$ alors $\S_f(u) = \S_f(t) - 1$. La mesure décroît donc strictement et \rightarrow_e est bien fortement normalisante. \square

Définition 9.17. Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . La **forme totalement expansée** d'un terme t est l'unique forme normale de t pour l'expansion.

9.2 Réductions source, intermédiaire, et cible

Dans le cas du λ -calcul la définition de la relation de réduction cible était relativement simple, et pouvait se ramener à une opération en deux étapes :

1. Expanser le supercombinateur du radical.
2. Appliquer la β -réduction usuelle.

Les règles plus riches des CRS ont pour première différence de pouvoir faire intervenir plusieurs fonctions du terme source (exemple 9.18), et demander ainsi l'expansion de plusieurs supercombinateurs. Nous pourrions ainsi prendre comme idée de base : expanser autant de supercombinateurs que nécessaire afin de pouvoir appliquer une règle du CRS d'origine, puis appliquer cette règle (puis d'autres ingrédients éventuels). Cependant, comme au chapitre 8 le but sera de définir toute cette opération comme un seul pas de réduction, de préférence décrit par une règle de réécriture.

Exemple 9.18.

Considérons des symboles \circ , f , g , et a d'arités respectives 2, 2, 2, et 0, et la règle CRS

$$\circ : \circ(\langle x \rangle Z_1(x), \langle y \rangle Z_2(y)) \rightarrow \langle x \rangle Z_1(Z_2(x))$$

Considérons un terme

$$t = \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), \langle y \rangle g(a, y))$$

Nous avons avec la règle \circ un pas de réduction

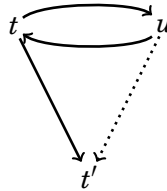
$$t \rightarrow \langle x \rangle f(g(a, x), \langle z \rangle f(z, g(a, x))) = t'$$

Notons $\psi^{-1}(\langle y \rangle g(\square, y)) = h_y$. Nous avons alors un pas de méta-lifting

$$t \rightarrow_{lft} \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), h_y(a)) = u$$

Nous pouvons encore obtenir une réduction vers t' en expansant d'abord h_y puis en appliquant la règle \circ . Nous pouvons également proposer une règle « directe » dans laquelle le symbole h_y est utilisé à gauche, et expansé à droite où cela est nécessaire.

$$\circ_y : \circ(\langle x \rangle Z_1(x), h_y(Z_0)) \rightarrow \langle x \rangle Z_1(g(Z_0, x))$$



Cette nouvelle règle \circ_y n'est plus une règle source, mais il ne s'agit pas non plus d'une règle cible : elle fait intervenir, à gauche comme à droite, des lieurs et des méta-variables d'arité non nulle, choses qui ne siéent guère à une règle du premier ordre. Voici donc une première différence avec le chapitre 8 précédent : il existera réellement une notion de réduction « intermédiaire », dont les règles ne seront ni celles de la réduction source ni celles de la réduction cible ¹.

Comment obtenir la règle \circ_y à partir de la règle source \circ ? Le membre gauche

$$\circ(\langle x \rangle Z_1(x), h_y(Z_0))$$

de la règle \circ_y peut être vu comme obtenu à partir du membre gauche

$$\circ(\langle x \rangle Z_1(x), \langle y \rangle Z_2(y))$$

de la règle \circ par contraction de l'abstraction $\langle y \rangle$. À un détail près : la contraction n'est pas définie sur des squelettes contenant des méta-variables d'arité non nulle comme $\langle y \rangle Z_2(y)$. Dans notre cas, ladite contraction n'a pas lieu directement dans le membre gauche de \circ , mais dans son raffinement

$$\circ(\langle x \rangle Z_1(x), \langle y \rangle g(Z_0, y))$$

obtenu avec la valuation partielle

$$\sigma = \{ Z_2 \mapsto \lambda x_1. g(Z_0, x_1) \}$$

Ainsi, le squelette $\langle y \rangle g(\square, y)$ ne contient plus de méta-variable et peut bien être contracté en h_y . Remarquons que cette valuation partielle σ assure également le passage entre les membres droits des règles \circ et \circ_y .

Pour obtenir les règles intermédiaires et cibles à partir des règles sources en utilisant la contraction, nous allons donc devoir considérer des raffinements des règles sources qui permettent de faire toutes les contractions voulues. Pour ceci, il faudra que les squelettes du membre gauche soient entièrement visibles et ne contiennent plus de méta-variables. Autrement dit, les raffinements canoniques sont de retour !

Ainsi, une règle intermédiaire κ_i sera obtenue en appliquant des contractions à une règle κ_R du raffinement canonique d'une règle κ du CRS d'origine.

Exemple 9.19.

Considérons la règle \circ_R suivante, issue du raffinement canonique de la règle \circ de l'exemple 9.18.

$$\circ_R : \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), \langle y \rangle g(Z_0, y)) \rightarrow \langle x \rangle f(g(Z_0, x), \langle z \rangle f(z, g(Z_0, x)))$$

En appliquant à l'envi aux membres gauche et droit de la règle \circ_R les contractions données par

$$\begin{aligned} \psi^{-1}(\langle y \rangle g(\square, y)) &= h_y \\ \psi^{-1}(\langle z \rangle f(z, \square)) &= h_z \\ \psi^{-1}(\langle x \rangle f(x, h_z(x))) &= h_x \end{aligned}$$

1. Ceci était en réalité déjà le cas au chapitre 8 mais avait été passé sous silence, avec des réductions cibles $@(f(t_1, \dots, t_n), a)$ dans lesquelles $\psi(f)$ contient des λ -abstractions internes.

nous obtenons par exemple, en plus de \circ_y , les règles suivantes dans lesquelles les symboles contractés sont soulignés.

$$\begin{aligned} \circ_z & : \circ(\langle x \rangle f(x, \underline{h_z}(x)), \langle y \rangle g(Z_0, y)) \rightarrow \langle x \rangle f(g(Z_0, x), \langle z \rangle f(z, (g(Z_0, x)))) \\ \circ_{xyz} & : \circ(\underline{h_x}, \underline{h_y}(Z_0)) \rightarrow \langle x \rangle f(g(Z_0, x), \underline{h_z}(g(Z_0, x))) \end{aligned}$$

Comme il faut pouvoir appliquer les règles intermédiaires à toute étape intermédiaire d'une transformation de *méta-lifting*, étant donnée une règle κ du CRS d'origine nous prévoirons au moins une règle intermédiaire pour chaque motif atteignable par contraction d'un membre gauche l_R d'une règle $\kappa_R : l_R \rightarrow r_R$ du raffinement canonique de κ . Ceci fait écho en particulier aux règles étiquetées (définition 4.42) et aux règles expansées (définition 7.12) pour lesquelles nous avons considéré tout étiquetage et toute expansion des membres gauches.

Il y a en revanche une question à trancher concernant la contraction éventuelle du membre droit : que décider pour les deux abstractions du membre droit $\langle x \rangle f(g(Z_0, x), \langle z \rangle f(z, g(Z_0, x)))$ de la règle \circ_R ? Écartons la possibilité de construire toutes les règles correspondant à toutes les contractions des membres gauche et droit, qui aurait un impact négatif sur l'éventuelle orthogonalité du système source. Nous cherchons plutôt une construction déterministe.

La question ne se posait pas dans le formalisme du λ -calcul. En effet, considérons dans le cadre du chapitre 8 précédent un pas de réduction intermédiaire

$$@ (f(t_1, \dots, t_n), u) \rightarrow s^{[x:=u]} [t_1, \dots, t_n]$$

Remarquons que toute λ -abstraction et tout symbole de supercombinateur du λ -terme étendu cible soit a un ancêtre, soit est introduit par $\psi(f) = \lambda x.s$. Sa forme contractée ou expansée est donc dictée par le λ -terme étendu source ou la définition de l'expandeur ψ .

Ce possible « héritage » par descendance de la forme contractée ou expansée d'un abstraction pourrait être mis en avant pour proposer que l'abstraction $\langle z \rangle$ du membre droit soit dans le même état que l'abstraction $\langle z \rangle$ du membre gauche. Officiellement la première ne descend pas de la seconde puisque, formellement, le raffinement efface cette information. Nous pourrions néanmoins nous rappeler que les deux occurrences sont issues d'une même valuation partielle

$$\sigma_R = \{ Z_1 \mapsto \underline{\lambda}x_1.f(x_1, \langle z \rangle f(z, x_1)), Z_2 \mapsto \underline{\lambda}x_1.g(Z_0, x_1) \}$$

Il serait même envisageable de régler la question de ces abstractions en considérant le raffinement canonique non par rapport à la signature \mathcal{S} d'origine, mais par rapport à la signature $\mathcal{S} \sqcup \mathcal{F}$ étendue, de sorte qu'une valuation partielle

$$\sigma_z = \{ Z_1 \mapsto \underline{\lambda}x_1.f(x_1, \underline{h_z}(x_1)), Z_2 \mapsto \underline{\lambda}x_1.g(Z_0, x_1) \}$$

génère directement à partir de la règle \circ de l'exemple 9.18 une règle raffinée

$$\circ(\langle x \rangle f(x, \underline{h_z}(x)), \langle y \rangle g(Z_0, y)) \rightarrow \langle x \rangle f(g(Z_0, x), \underline{h_z}(g(Z_0, x)))$$

dans laquelle la contraction de l'abstraction $\langle z \rangle$ est déjà prise en compte (et ce de manière uniforme à gauche et à droite).

Cependant, cette solution ne serait pas suffisante : dans une règle CRS comme la règle \circ de l'exemple 9.18, des abstractions apparaissent à gauche comme à droite indépendamment de tout raffinement. D'une part il faut donc explicitement prévoir la possible contraction des abstractions du membre gauche. D'autre part, les abstractions du membre droit n'étant pas reliées aux abstractions du membre gauche, nous ne pouvons nous baser sur un argument même informel de descendance pour décider de leur état. En particulier, la décision de contracter ou non l'abstraction $\langle x \rangle$ dans le membre droit $\langle x \rangle Z_1(Z_2(x))$ relève de l'arbitraire.

Seul critère restant : que les règles cibles, qui seront déduites des règles intermédiaires dans les cas où les termes sont des formes normales du *méta-lifting*, soient bien des règles du premier ordre, dans lesquelles aucun lieu n'apparaît. Solution retenue : prendre la forme normale par *méta-lifting* du membre droit de la règle raffinée considérée.

Exemple 9.20.

Considérons la règle raffinée \circ_R de l'exemple 9.19

$$\circ_R : \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), \langle y \rangle g(Z_0, y)) \rightarrow \langle x \rangle f(g(Z_0, x), \langle z \rangle f(z, g(Z_0, x)))$$

Notons

$$\begin{aligned} \psi^{-1}(\langle z \rangle f(z, \square)) &= h_z \\ \psi^{-1}(\langle x \rangle f(g(\square, x), h_z(g(\square, x)))) &= h_x^R \end{aligned}$$

Alors toutes les règles intermédiaires issues de la règle raffinée \circ_R auront pour membre droit

$$h_x^R(Z_0, Z_0)$$

Pour une définition formelle nous utilisons la notion suivante de saturation d'une règle.

Définition 9.21 (Saturation d'une règle). Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de *lifting* stable pour Σ . Soit $\kappa : l \rightarrow r$ une règle de $R(\mathcal{R})$. La **saturation** $S(\kappa)$ de la règle $\kappa : l \rightarrow r$ est l'ensemble de règles défini par

$$S(\kappa) = \{l' \rightarrow \text{lft}(r) \mid l \rightarrow_{\text{lft}} l'\}$$

où nous rappelons par convention 9.15 que $\text{lft}(r)$ désigne la forme normale de r par *méta-lifting*.

Exemple 9.22.

Considérons la règle raffinée \circ_R des exemples 9.19 et 9.20

$$\circ_R : \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), \langle y \rangle g(Z_0, y)) \rightarrow \langle x \rangle f(g(Z_0, x), \langle z \rangle f(z, g(Z_0, x)))$$

Notons

$$\begin{aligned}
\psi^{-1}(\langle y \rangle g(\square, y)) &= h_y \\
\psi^{-1}(\langle z \rangle f(z, \square)) &= h_z \\
\psi^{-1}(\langle x \rangle f(x, h_z(x))) &= h_x \\
\psi^{-1}(\langle x \rangle f(x, \langle z \rangle f(z, x))) &= h'_x \\
\psi^{-1}(\langle x \rangle f(g(\square, x), h_z(g(\square, x)))) &= h_x^R
\end{aligned}$$

La saturation de la règle \circ_R est alors l'ensemble suivant, composé de huit règles correspondant à tous les états contractés ou non-contractés des abstractions $\langle y \rangle$, $\langle z \rangle$, et $\langle x \rangle$.

$$\begin{aligned}
S(\circ_R) = \{ & \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), \langle y \rangle g(Z_0, y)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(\langle x \rangle f(x, h_z(x)), \langle y \rangle g(Z_0, y)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(h_x, \langle y \rangle g(Z_0, y)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(h'_x, \langle y \rangle g(Z_0, y)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(\langle x \rangle f(x, \langle z \rangle f(z, x)), h_y(Z_0)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(\langle x \rangle f(x, h_z(x)), h_y(Z_0)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(h_x, h_y(Z_0)) \rightarrow h_x^R(Z_0, Z_0) \ , \\
& \circ(h'_x, h_y(Z_0)) \rightarrow h_x^R(Z_0, Z_0) \ }
\end{aligned}$$

Remarque 9.23. *Tout pas de réduction par une règle saturée se décompose en une séquence de réduction $\rightarrow_e \cdot \rightarrow_s \cdot \rightarrow_c$ où \rightarrow_s est un pas de réduction par la règle d'origine.*

Remarquons qu'à l'instar des règles étiquetées (définition 4.42) et des règles expansées (définition 7.12), les règles saturées associées à une règle $l \rightarrow r$ balayent toutes les configurations possibles du membre gauche mais ont toutes le même membre droit.

Définition 9.24 (Réductions source, intermédiaire, et cible). *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Les **règles sources** sont les règles de \mathcal{R} . Les **règles intermédiaires** sont données par la saturation du raffinement canonique de \mathcal{R} : $S(R(\mathcal{R}))$. Les **règles cibles** sont les règles intermédiaires dont le membre gauche est en forme normale pour le méta-lifting \rightarrow_{lf} . Un pas de réduction source (resp. intermédiaire, cible) est l'application d'une règle source (resp. intermédiaire, cible) à une position non gelée, et est noté \rightarrow_s (resp. \rightarrow_i , \rightarrow_f).*

Insistons sur le fait que les réductions source, intermédiaire, et cible sont soumises à la restriction de réduction faible.

Convention 9.25. *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS, (\mathcal{F}, ψ) un système de lifting stable pour Σ , et $R(\mathcal{R})$ le raffinement canonique de \mathcal{R} . Remarquons que par lemme 9.14, chaque règle $\kappa : l \rightarrow r$ dans $R(\mathcal{R})$ définit une unique règle cible, que nous noterons $\text{lft}(\kappa) : \text{lft}(l) \rightarrow \text{lft}(r)$.*

Lemme 9.26 (Source/intermédiaire). *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS, (\mathcal{F}, ψ) un système de lifting stable pour Σ , et t un terme sur \mathcal{S} .*

$$\begin{array}{l}
- \text{ Si } \begin{array}{c} t \\ \downarrow s \\ t' \end{array} \text{ alors il existe } u' \text{ tel que } \begin{array}{c} t \\ \searrow i \\ t' \xrightarrow{\text{lift}} u' \end{array} . \\
- \text{ Si } \begin{array}{c} t \\ \searrow i \\ u' \end{array} \text{ alors il existe } t' \text{ tel que } \begin{array}{c} t \\ \downarrow s \\ t' \xrightarrow{\text{lift}} u' \end{array} .
\end{array}$$

Démonstration. Supposons $t \rightarrow_s t'$. Alors $t = c[l^\sigma] \rightarrow_s c[r^\sigma] = t'$ par une règle raffinée $\kappa : l \rightarrow r$. Alors $t = c[l^\sigma] \rightarrow_s c[\text{lift}(r)^\sigma]$ par la règle intermédiaire $l \rightarrow \text{lift}(r)$, et $t' = c[r^\sigma] \rightarrow_{\text{lift}} c[\text{lift}(r)^\sigma]$.

Supposons $t \rightarrow_i u'$. Alors par remarque 9.23 de décomposition des pas de réduction intermédiaire, $t \rightarrow_e \rightarrow_s \rightarrow_c u'$. Or t est un terme sur \mathcal{S} , donc une forme normale pour \rightarrow_e . Donc $t \rightarrow_s \rightarrow_c u'$. \square

Dans la section 9.3 suivante nous allons montrer qu'étant donné un CRS faible Σ , tout système de *lifting* stable pour Σ génère un système cible qui est bisimilaire à Σ . Comme à la section 8.3 pour le λ -calcul faible, cette démonstration sera basée sur un lemme de commutation de la réduction intermédiaire avec le *méta-lifting*.

9.3 Bisimulation

Le cœur du résultat de bisimulation est le lemme 9.31 de commutation entre le *méta-lifting* et la réduction intermédiaire, qui sera prouvé à l'aide d'une part des lemmes mineurs 9.28 à 9.30 et d'autre part grâce à la décomposition du *méta-lifting* sur la contraction \rightarrow_c et l'expansion \rightarrow_e (lemme 9.13) et aux propriétés de réciprocity de \rightarrow_c et \rightarrow_e . L'inspection de la preuve du lemme 9.31 de commutation permettra également de conclure cette section en vérifiant que la transformation de *lifting* préserve l'éventuelle orthogonalité des systèmes (théorème 9.37).

Convention 9.27. Soit un substitut $\underline{\lambda}x_1 \dots x_n.t$. Nous notons $\underline{\lambda}x_1 \dots x_n.t \rightarrow \underline{\lambda}x_1 \dots x_n.t'$ si $t \rightarrow t'$. Soit une valuation σ . Nous notons $\sigma \rightarrow \sigma'$ s'il existe une méta-variable Z_0 telle que $\sigma(Z_0) \rightarrow \sigma'(Z_0)$ et pour toute méta-variable $Z \neq Z_0$, $\sigma(Z) = \sigma'(Z)$.

Lemme 9.28. Soit t un méta-terme et σ une valuation sûre pour t . Supposons $\sigma \rightarrow \sigma'$. Alors σ' est sûre pour t .

Démonstration. Il existe un substitut $\underline{\lambda}x_1 \dots x_n.u$ dans le codomaine de σ tel que $u \rightarrow u'$ et le substitut $\underline{\lambda}x_1 \dots x_n.u'$ est dans le codomaine de σ' . Comme la valuation σ est sûre pour t et $\text{fv}(u') \subseteq \text{fv}(u)$, la valuation σ' est encore sûre pour t . \square

Lemme 9.29. Soit t un méta-terme et σ une valuation sûre pour t . Supposons $\sigma \rightarrow \sigma'$. Alors $t^\sigma \rightarrow t^{\sigma'}$. Si de plus t est linéaire et ne contient que des méta-variables d'arité 0, alors $t^\sigma \rightarrow t^{\sigma'}$.

Démonstration. Par induction sur t . □

Lemme 9.30. *Considérons un pas de réduction $t \rightarrow_{lft} u$. Alors $\text{fv}(t) = \text{fv}(u)$.*

Démonstration. Par lemme 9.13 nous ramenons la preuve à la vérification de la propriété pour les réductions \rightarrow_c de contraction et \rightarrow_e d'expansion.

Supposons $t \rightarrow_c u$. Alors $\text{fv}(u) \subseteq \text{fv}(t)$. Or, par lemme 9.7 nous avons également $u \rightarrow_e t$, d'où $\text{fv}(t) \subseteq \text{fv}(u)$, et finalement $\text{fv}(t) = \text{fv}(u)$.

De même, si $u \rightarrow_e t$ alors $\text{fv}(u) = \text{fv}(t)$. □

Nous pouvons maintenant prouver le résultat de commutation entre le *méta-lifting* et la réduction intermédiaire. Notons que la preuve de ce lemme 9.31 est plus courte que celle du lemme 8.32 alors même que cette dernière ne traitait qu'un seul sens : l'uniformité du formalisme des CRS permet de gagner en concision dans les analyses de cas.

Lemme 9.31 (Commutation). *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ .*

$$\begin{array}{l}
 - \text{ Si } \begin{array}{ccc} t & \xrightarrow{lft} & u \\ i \downarrow & & \\ t' & & \end{array} \text{ alors il existe } u' \text{ tel que } \begin{array}{ccc} & & u \\ & & \downarrow i \\ t' & \xrightarrow{lft} & u' \end{array} . \\
 - \text{ Si } \begin{array}{ccc} t & \xrightarrow{lft} & u \\ & & \downarrow i \\ & & u' \end{array} \text{ alors il existe } t' \text{ tel que } \begin{array}{ccc} & & t \\ i \downarrow & & \\ t' & \xrightarrow{lft} & u' \end{array} .
 \end{array}$$

Démonstration. Supposons $t'_i \leftarrow t \rightarrow_{lft} u$. Notons c et $l \rightarrow r$ le contexte et la règle de la réduction $t = c[l^\sigma] \rightarrow_i c[r^\sigma] = t'$. Par définition de la saturation, il existe $l_0 \rightarrow r_0$ dans le raffinement canonique du CRS d'origine telle que $l_0 \rightarrow_{lft} l$ et $r = \text{lft}(r_0)$. Cas sur les positions relatives des deux radicaux.

I) Réductions à des positions disjointes. Ce cas est immédiat.

II) Réduction \rightarrow_{lft} dans le motif l du radical \rightarrow_i . Dans ce cas $u = c[l'^\sigma]$ avec $l \rightarrow_{lft} l'$. En particulier $l_0 \rightarrow_{lft} l'$. Par saturation $u = c[l'^\sigma] \rightarrow_i c[\text{lft}(r_0)^\sigma] = c[r^\sigma] = t'$.

III) Réduction \rightarrow_{lft} dans un paramètre du radical \rightarrow_i . Dans ce cas $u = c[l^{\sigma'}]$ avec $\sigma \rightarrow_{lft} \sigma'$, et par lemme 9.28 $u = c[l^{\sigma'}] \rightarrow_i c[r^{\sigma'}]$ et nous notons $u' = c[r^{\sigma'}]$. Par lemme 9.29 $t' = c[r^\sigma] \rightarrow_{lft} c[r^{\sigma'}] = u'$.

IV) Réduction \rightarrow_i dans le motif du radical \rightarrow_{lft} . Ce cas est interdit par la restriction faible.

V) Réduction \rightarrow_i dans un paramètre du radical \rightarrow_{lft} . Dans ce cas $t = c_l[l_l^{\sigma'}] \rightarrow_{lft} c_l[r_l^{\sigma'}] = u$, avec $\sigma_l \rightarrow_i \sigma'_l$. Alors par lemme 9.29 et en remarquant que l_l et r_l sont linéaires et ne contiennent que des méta-variables d'arité 0 nous avons $t = c_l[l_l^{\sigma'}] \rightarrow_i c_l[l_l^{\sigma'_l}] = t'$ et $u = c_l[r_l^{\sigma'}] \rightarrow_i c_l[r_l^{\sigma'_l}]$. Notons $u' = c_l[r_l^{\sigma'_l}]$. Par lemme 9.28 $t' = c_l[l_l^{\sigma'_l}] \rightarrow_{lft} c_l[r_l^{\sigma'_l}] = u'$.

Supposons $t \rightarrow_{\text{lft}} u \rightarrow_i u'$. Notons c et $l \rightarrow r$ le contexte et la règle de la réduction $u = c[l^\sigma] \rightarrow_i c[r^\sigma] = u'$. Par définition de la saturation, il existe $l_0 \rightarrow r_0$ dans le raffinement canonique du CRS d'origine telle que $l_0 \rightarrow_{\text{lft}} l$ et $r = \text{lft}(r_0)$. Notons c_l et $l_l \rightarrow r_l$ le contexte et la règle de la réduction $u = c_l[l_l^{\sigma_l}] \rightarrow_i c_l[r_l^{\sigma_l}] = u'$. Cas sur les positions relatives de l et r_l .

- I) Positions disjointes. Ce cas est immédiat.
- II) Réduit r_l dans le motif l du radical \rightarrow_i . Dans ce cas $t = c[l^\sigma] \rightarrow_{\text{lft}} c[l^\sigma] = u$ avec $l' \rightarrow_{\text{lft}} l$. Rappelons que $l_0 \rightarrow_{\text{lft}} l$. Donc par lemmes 9.13 de décomposition du méta-lifting sur \rightarrow_c et \rightarrow_e et 9.7 de réciprocity entre \rightarrow_c et \rightarrow_e , les motifs l' et l_0 sont \rightarrow_e -convertibles. Donc par lemme 9.16 de convergence l' et l_0 ont même forme normale pour \rightarrow_e . Or l_0 est une forme normale pour \rightarrow_e , donc $l' \rightarrow_e l_0$. Ainsi par lemme 9.7 de réciprocity entre \rightarrow_c et \rightarrow_e nous avons $l_0 \rightarrow_c l'$, d'où $l_0 \rightarrow_{\text{lft}} l'$. Donc par saturation $t = c[l^\sigma] \rightarrow_i c[\text{lft}(r_0)^\sigma] = c[r^\sigma] = u'$.
- III) Réduit r_l dans un paramètre du radical \rightarrow_i . Dans ce cas $t = c[l^{\sigma'}] \rightarrow_{\text{lft}} c[l^\sigma] = u$ avec par lemme 9.30 σ' une valuation sûre. En particulier $t = c[l^{\sigma'}] \rightarrow_i c[r^{\sigma'}]$. Notons $t' = c[r^{\sigma'}]$. Par lemme 9.29 $t' = c[r^{\sigma'}] \rightarrow_{\text{lft}} c[r^\sigma] = u'$.
- IV) Réduction \rightarrow_i dans le réduit r_l du radical \rightarrow_{lft} . Ce cas est impossible car $r_l = f \in \mathcal{F}$.
- V) Réduction \rightarrow_i dans un paramètre du réduit du radical \rightarrow_{lft} . Dans ce cas $u = c_l[r_l^{\sigma_l}] \rightarrow_i c_l[r_l^{\sigma'_l}] = u'$, avec $\sigma_l \rightarrow_i \sigma'_l$. Alors par lemme 9.29 et car l_l est linéaire et ne contient que des méta-variables d'arité 0, nous avons $u = c_l[l_l^{\sigma'_l}] \rightarrow_i c_l[l_l^{\sigma'_l}]$. Notons $t' = c_l[l_l^{\sigma'_l}]$. Par lemme 9.28 $t' \rightarrow_{\text{lft}} u'$. \square

Remarque 9.32. Nous pouvons noter que les deux parties de la preuve sont essentiellement symétriques l'une de l'autre, excepté pour le point II) concernant une opération de méta-lifting à l'intérieur du motif l de la règle $\kappa : l \rightarrow r$ du pas de réduction intermédiaire. Dans le cas $l \rightarrow_{\text{lft}} l'$ il est évident que $l' \rightarrow r$ est, comme κ , une règle de la saturation de la règle source raffinée $l_0 \rightarrow r_0$. Dans le cas $l' \rightarrow_{\text{lft}} l$ en revanche il faut démontrer que nous avons $l_0 \rightarrow_{\text{lft}} l'$, à l'aide des propriétés de convergence de l'expansion \rightarrow_e .

Remarque 9.33. Dans les deux propositions du lemme 9.31, les deux pas de réduction $t \rightarrow_i t'$ et $u \rightarrow_i u'$ utilisent des règles de réécriture $\kappa_t : l_t \rightarrow r_t$ et $\kappa_u : l_u \rightarrow r_u$ qui sont dans la saturation de la même règle raffinée $\kappa_0 : l_0 \rightarrow r_0$. Plus précisément, dans les points I), III), IV), et V) les règles κ_t et κ_u sont égales, et dans le point II) nous avons $l_t \rightarrow_{\text{lft}} l_u$.

Pour un énoncé complet de bisimulation, qui par définition 0.15 implique une relation sur les objets (les termes) et une relation sur les pas de réduction, il nous faut encore étendre la fonction de *lifting* $\text{lft}(\cdot)$ aux pas de réduction.

Convention 9.34. Soit $\rho_0 : t = c[l_0^{\sigma_0}] \rightarrow_s c[r_0^{\sigma_0}] = t'$ un pas de réduction par une règle source $\kappa_0 : l_0 \rightarrow r_0$. Par lemme 4.30 de factorisation canonique il existe une unique règle $\kappa : l \rightarrow r$ dans le raffinement canonique $R(\kappa_0)$ de la règle source

κ_0 telle que nous avons un pas de réduction $\rho : t = c[l^\sigma] \rightarrow_s c[r^\sigma] = t'$. Alors par lemme 9.26 nous avons un pas de réduction intermédiaire $\rho_i : t = c[l^\sigma] \rightarrow_i c[\text{lft}(r)^\sigma]$, et par lemme 9.31 nous avons la relation de réduction $\text{lft}(t) \rightarrow_f \text{lft}(t')$. De plus, par remarque 9.33 nous avons plus précisément un pas de réduction $\rho_f : \text{lft}(t) \rightarrow_f \text{lft}(t')$ qui utilise la règle $\text{lft}(\kappa)$. Nous étendons la fonction $\text{lft}(\cdot)$ aux pas de réduction en notant $\text{lft}(\rho_0) = \rho_f$.

Le lemme 9.31 de commutation associé avec le lemme 9.26 de lien entre la réduction source et la réduction intermédiaire a pour corollaire immédiat le théorème de bisimulation suivant.

Théorème 9.35. *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS et (\mathcal{F}, ψ) un système de lifting stable pour Σ . La fonction $\text{lft}(\cdot)$ de lifting est une bisimulation entre la réduction source et la réduction cible.*

Nous allons maintenant vérifier que cette fonction de *lifting*, en plus d'être une bisimulation préserve l'éventuelle orthogonalité des systèmes dans le sens suivant : en prenant pour source un CRS orthogonal nous obtenons pour cible un TRS qui est lui aussi orthogonal (théorème 9.37). Ce résultat est obtenu en constatant que les règles intermédiaires forment un système orthogonal en réduction faible (lemme 9.36), ce qui peut être fait à partir du lemme 9.31 de commutation.

Lemme 9.36 (Orthogonalité en réduction faible). *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS orthogonal et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Alors les règles intermédiaires forment un CRS orthogonal en réduction faible.*

Démonstration. Vérifions d'abord la linéarité gauche. Soit une règle intermédiaire $\kappa : l \rightarrow r$. Par définition il existe une règle $\kappa_0 : l_0 \rightarrow r_0$ dans le raffinement canonique d'une règle de \mathcal{R} telle que $l_0 \rightarrow_{\text{lft}} l$. Or $R(\mathcal{R})$ est orthogonal en réduction faible, par lemme 4.34. En particulier le motif l_0 est linéaire gauche. De plus \rightarrow_{lft} n'opère aucune duplication. Donc par récurrence sur la longueur de la séquence $l_0 \rightarrow_{\text{lft}} l$ le motif l est linéaire gauche.

Vérifions maintenant que les règles intermédiaires sont sans recouvrement en réduction faible. Considérons un radical l_o^σ pour une règle intermédiaire $\kappa_o : l_o \rightarrow r_o$ contenant un sous-radical $l_i^{\sigma_i}$ pour une règle intermédiaire $\kappa_i : l_i \rightarrow r_i$ telle que $\kappa_o \neq \kappa_i$. Notons $l_o^{\sigma_o} = c_i[l_i^{\sigma_i}]$. Supposons que $l_i^{\sigma_i}$ n'est pas contenu dans σ_o ni n'apparaît à une position gelée de l_o (faute de quoi il n'y aurait rien à vérifier). Nous avons donc deux pas de réduction intermédiaire $\rho_o : l_o^{\sigma_o} \rightarrow r_o^{\sigma_o}$ et $\rho_i : c_i[l_i^{\sigma_i}] \rightarrow c_i[r_i^{\sigma_i}]$. Enfin, par définition de la saturation il existe deux règles $\kappa'_o : l'_o \rightarrow r'_o$ et $\kappa'_i : l'_i \rightarrow r'_i$ dans $R(\mathcal{R})$ telles que $l'_o \rightarrow_c l_o$, $l'_i \rightarrow_c l_i$, $r_o = \text{lft}(r'_o)$, et $r_i = \text{lft}(r'_i)$.

Notons t la forme totalement expansée de $l_o^{\sigma_o}$. Par définition $l_o^{\sigma_o} \rightarrow_e t$, et par réciprocity entre \rightarrow_e et \rightarrow_c (lemme 9.7) nous avons également $t \rightarrow_c l_o^{\sigma_o}$ et donc $t \rightarrow_{\text{lft}} l_o^{\sigma_o}$. Par lemme 9.31 nous avons donc deux pas de réduction intermédiaire $\rho'_o : t = (l'_o)^{\sigma'_o} \rightarrow t'_o$ et $\rho'_i : t = c'_i[(l'_i)^{\sigma'_i}] \rightarrow t'_i$ avec les règles intermédiaires $l'_o \rightarrow \text{lft}(r'_o)$ et $l'_i \rightarrow \text{lft}(r'_i)$ (qui sont respectivement dans les saturations de $\kappa'_o : l'_o \rightarrow$

r'_o et $\kappa'_i : l'_i \rightarrow r'_i$). De plus, par inspection de la preuve du lemme 9.31, comme $l_i^{\sigma_i}$ n'est pas contenu dans σ_o alors $(l'_i)^{\sigma'_i}$ n'est pas contenu dans σ'_o , et comme la position de $l_i^{\sigma_i}$ n'est pas une position gelée de l_o alors la position de $(l'_i)^{\sigma'_i}$ n'est pas une position gelée de l'_o . Donc le radical $(l'_o)^{\sigma'_o}$ et son sous-radical $(l'_i)^{\sigma'_i}$ forment un recouvrement des règles intermédiaires $l'_o \rightarrow \text{ft}(r'_o)$ et $l'_i \rightarrow \text{ft}(r'_i)$.

De plus, $(l'_o)^{\sigma'_o}$ et $(l'_i)^{\sigma'_i}$ sont encore des radicaux pour les règles sources κ'_o et κ'_i , qui forment un recouvrement dans $R(\mathcal{R})$, ce qui contredit l'orthogonalité en réduction faible de $R(\mathcal{R})$ (lemme 4.34).

De même si $\kappa_i = \kappa_o$ et $c_i \neq \square$, d'où nous déduisons que $S(R(\mathcal{R}))$ est sans recouvrement en réduction faible. Ainsi la réduction intermédiaire de notre système est orthogonale en réduction faible. \square

L'orthogonalité en réduction faible du lemme 9.36 s'applique en particulier au sous-ensemble formé par les règles cibles. Or les règles cibles sont des règles du premier ordre, qui ne contiennent aucun lieu et donc aucun squelette ni aucune position gelée. Sur de telles règles l'orthogonalité en réduction faible est confondue avec l'orthogonalité « ordinaire ». Et nous avons ainsi en corollaire immédiat le théorème 9.37 d'orthogonalité du système cible.

Théorème 9.37 (Orthogonalité du système cible). *Soient $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS orthogonal et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Alors les règles cibles forment un TRS orthogonal.*

9.4 Application à un système étiqueté

Nous avons présenté dans les sections précédentes une notion de *lifting* généralisée à n'importe quel CRS, et prouvé qu'elle définissait une bisimulation entre un CRS faible et un système de réécriture du premier ordre. Dans le chapitre 8 sur le λ -calcul, l'étape suivante consistait à définir une nouvelle notion de λ -*lifting* étiqueté et à prouver encore une fois les mêmes résultats dans le cadre étiqueté. Cette deuxième étape n'a pas lieu d'être ici, puisqu'au chapitre 4 pour l'étiquetage pleinement paresseux comme au chapitre 7 pour l'étiquetage causal nous avons toujours pris soin d'« internaliser » les mécanismes d'étiquetages et ainsi de faire du CRS étiqueté un CRS « ordinaire ».

Il suffit donc pour obtenir un *lifting* étiqueté d'exhiber un système de *lifting* stable pour le CRS représentant l'étiquetage de notre système d'origine. Nous allons dans cette section appliquer ce principe à l'étiquetage causal défini au chapitre 7, en prévision du théorème d'optimalité auquel nous aboutirons au chapitre 10 suivant (rappelons au passage que par théorèmes 7.29 et 7.36 les étiquetages des chapitres 4 et 7 définissent la même réduction partagée).

Après définition de ce système de *lifting* stable pour le système étiqueté à la section 9.4.1, nous nous servirons des étiquettes pour établir deux nouvelles propriétés du *lifting* : nous verrons à la section 9.4.2 que le *lifting* préserve les notions de descendance, de résiduation et de nécessité, puis à la section 9.4.3

que le *lifting* définit une bisimulation entre les réductions partagées de la source et de la cible.

9.4.1 Définition d'un système de *lifting* étiqueté stable.

Ainsi, considérons un CRS $\Sigma = (\mathcal{S}, \mathcal{R})$ et son étiquetage causal $\Sigma^{\mathcal{L}} = (\mathcal{S}^{\mathcal{L}}, \mathcal{R}^{\mathcal{L}})$ selon le chapitre 7 (l'ensemble de règles $\mathcal{R}^{\mathcal{L}}$ est en réalité de la forme $R(E(\mathcal{R}))^{\mathcal{L}}$ mais nous garderons ici la forme abrégée de la notation). Supposons avoir un système de *lifting* (\mathcal{F}, ψ) stable pour Σ . Nous allons en déduire un système de *lifting* stable pour $\Sigma^{\mathcal{L}}$.

Pour commencer, de même qu'à la section 8.5 (définition 8.41) nous allons étendre la notion d'étiquetage initial à la signature étendue $\mathcal{S} \sqcup \mathcal{F}$ d'une manière qui distingue les symboles de la signature d'origine \mathcal{S} des symboles de supercombinateurs apportés par \mathcal{F} . Le but de cette distinction est de tenir compte des positions qui sont « cachées » dans les supercombinateurs et de donner à chaque symbole d'un terme t non sa position dans t mais la position dite *expansée* qu'il aurait dans la forme totalement expansée de t .

Cet étiquetage initial (donné à la définition 9.39) a donc deux caractéristiques : sur les termes sources il coïncide avec l'étiquetage initial selon la définition 7.17, et sur les termes étendus il dépend de la notion de *position expansée* (définition 9.38) qui elle-même est paramétrée par le système de *lifting* (\mathcal{F}, ψ) considéré.

Définition 9.38 (Positions expansées). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . Soient t un méta-terme sur $\mathcal{S} \sqcup \mathcal{F} \sqcup \{\square\}$ et $p \in \text{pos}(t)$ une position de t . L'**expansion** de p relativement à t et ψ est la position $\psi_t(p)$ définie par :*

$$\begin{aligned} \psi_t(\epsilon) &= \epsilon \\ \psi_{\langle x \rangle t}(1 \cdot p) &= 1 \cdot \psi_t(p) \\ \psi_{f(t_1, \dots, t_n)}(i \cdot p) &= i \cdot \psi_{t_i}(p) & f \in \mathcal{S}, i \in \{1, \dots, n\} \\ \psi_{Z(t_1, \dots, t_n)}(i \cdot p) &= i \cdot \psi_{t_i}(p) & i \in \{1, \dots, n\} \\ \psi_{f(t_1, \dots, t_n)}(i \cdot p) &= \psi_{\psi(f)}(q_i) \cdot \psi_{t_i}(p) & f \in \mathcal{F}, i \in \{1, \dots, n\}, q_i \text{ position du } i^{\text{ème}} \text{ trou de } \psi(f) \end{aligned}$$

Remarquons que cette position est bien définie car ψ est bien fondé.

Définition 9.39 (Étiquetage initial). *Soient \mathcal{S} une signature et (\mathcal{F}, ψ) un système de lifting pour \mathcal{S} . Pour toute position p la fonction $i_p(\cdot)$ transforme un terme sur $\mathcal{S} \sqcup \mathcal{F}$ en un terme sur $(\mathcal{S} \sqcup \mathcal{F})^{\mathcal{L}}$, avec les règles suivantes :*

$$\begin{aligned} i_p(x) &= x \\ i_p([x]t) &= [x]i_p(t) \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p \cdot 1}(t_1), \dots, i_{p \cdot n}(t_n)) & f \in \mathcal{S} \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p \cdot q_1}(t_1), \dots, i_{p \cdot q_n}(t_n)) & f \in \mathcal{F}, q_i = i^{f(t_1, \dots, t_n), \psi} \end{aligned}$$

Soit t un terme sur $\mathcal{S} \sqcup \mathcal{F}$. L'**étiquetage initial** de t est le terme $i_\epsilon(t)$ (sur $(\mathcal{S} \sqcup \mathcal{F})^{\mathcal{L}}$), que nous notons encore $i(t)$.

Rappelons que par convention 0.57 nous avons l'identité suivante :

$$i_p(\langle x \rangle t) = i_p(\diamond([x]t)) = \diamond^p(i_{p-1}([x]t)) = \diamond^p([x]i_{p-1}(t)) = \langle x \rangle^p i_{p-1}(t)$$

Nous pouvons maintenant déduire un système de lifting stable pour $\Sigma^{\mathcal{L}}$ en intégrant l'étiquetage initial de la définition 9.39 à l'expansor non étiqueté ψ .

Définition 9.40. *Considérons l'étiquetage $\mathcal{F}^{\mathcal{L}} = \{ f^\alpha \mid f \in \mathcal{F}, \alpha \in \mathcal{L} \}$ de la signature \mathcal{F} . Soit de plus un symbole étiqueté $f^\alpha \in \mathcal{F}^{\mathcal{L}}$ avec $\alpha = [\Omega_n, \dots, [\Omega_1, p]]$. Alors définissons*

$$\psi^{\mathcal{L}}(f^\alpha) = [\Omega_n, \dots, [\Omega_1, i_p(\psi(f))]]$$

Remarque 9.41. *Nous avons dans tous les cas $\tau(\psi^{\mathcal{L}}(f^\alpha)) = \alpha$.*

Lemme 9.42. *Le système de lifting $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ est stable pour $\Sigma^{\mathcal{L}}$.*

Démonstration. Le codomaine de $\psi^{\mathcal{L}}$ est l'ensemble des squelettes de la forme $[\Omega_n, \dots, [\Omega_1, i_p(s)]]$ où $\Omega_1, \dots, \Omega_n$ sont des séquences d'étiquettes, p est une position, et le squelette s est dans le codomaine de ψ . Comme par hypothèse (\mathcal{F}, ψ) est un système stable, nous savons que la réduction de Σ , la ψ -contraction et la ψ -expansion préservent le codomaine de ψ . Il ne reste donc à vérifier que la forme de l'étiquetage.

Considérons un terme t sur $(\mathcal{S} \sqcup \mathcal{F})^{\mathcal{L}}$ vérifiant la propriété de contractabilité $C_{\psi^{\mathcal{L}}}(t)$ et regardons les trois stabilités :

- I) Supposons $t = c[l^\sigma] \rightarrow c[r^\sigma] = t'$ par une règle $l \rightarrow r$ de $\mathcal{R}^{\mathcal{L}}$ et considérons un squelette s dans t' . Raisonnons par cas sur l'intersection de s avec c , r et σ .
- i) Si s est contenu dans c , alors s est un squelette de t et par hypothèse s est dans le codomaine de $\psi^{\mathcal{L}}$.
 - ii) Si s est intersecte à la fois c et r^σ , alors il existe une variable x liée par c telle que $x \in \text{fv}(r^\sigma)$. En particulier $x \in \text{fv}(r) \cup \text{fv}(\sigma)$. Or par définition le membre droit de règle r est un méta-terme clos. Donc $x \in \text{fv}(\sigma)$, ce qui contredit la réduction faible $t \rightarrow t'$.
 - iii) Si s est contenu dans r , alors il existe une position p (telle que $r|_p = s$) et un squelette s^* sur \mathcal{S} tels que $s = i_p(s^*)$. Remarquons de plus que le squelette s^* apparaît dans la règle $l^\bullet \rightarrow r^\bullet$ de \mathcal{R} . Donc par hypothèse sur ψ le squelette s^* est dans le codomaine de ψ . Donc $s = i_p(s^*)$ est dans le codomaine de $\psi^{\mathcal{L}}$.
 - iv) Si s intersecte à la fois r et σ , alors il existe une variable x liée dans r et libre dans le codomaine de σ , ce qui contredit la sûreté de la valuation σ pour la règle $l \rightarrow r$.
 - v) Si s est contenu dans le codomaine de σ , alors s est un squelette de t et par hypothèse s est dans le codomaine de $\psi^{\mathcal{L}}$.
- II) Supposons $t = c[\langle x \rangle^\alpha u[t_1, \dots, t_n]] \rightarrow_c c[f^\alpha(t_1, \dots, t_n)] = t'$ par $\psi^{\mathcal{L}}$ -contraction et considérons un squelette s dans t' . Raisonnons par cas sur l'intersection de s avec c , f^α et les t_1, \dots, t_n .

- i) Si s est contenu dans c ou dans l'un des t_1, \dots, t_n , alors s est un squelette de t et pas hypothèse s est dans le codomaine de $\psi^{\mathcal{L}}$.
- ii) Si s contient f^α , alors notons $s = c^s[f^\alpha(t_1^s, \dots, t_n^s)]$. En particulier les t_i^s sont libres dans $\langle x \rangle^\alpha u[t_1^s, \dots, t_n^s]$ et il existe une variable y libre dans un t_i^s et liée dans c^s . Alors l'ancêtre de s dans t est $s' = c^s[\langle x \rangle^\alpha u[t_1^s, \dots, t_n^s]]$, et par hypothèse s' est dans le codomaine de $\psi^{\mathcal{L}}$. En particulier, en notant $\alpha = [\Omega_n, \dots, [\Omega_1, p]]$, nous avons

$$\langle x \rangle^\alpha u[t_1^s, \dots, t_n^s] = [\Omega_n, \dots, [\Omega_1, i_p(\langle x \rangle u^*[(t_1^s)^\bullet, \dots, (t_n^s)^\bullet])]]$$

Donc pour tout i , en notant q_i la position du i -ième trou de $\langle x \rangle^\alpha u$, nous avons $t_i^s = [\Omega_n, \dots, [\Omega_1, i_{p \cdot q_i}((t_i^s)^\bullet)]]$, d'où

$$f^\alpha(t_1^s, \dots, t_n^s) = [\Omega_n, \dots, [\Omega_1, i_p(f((t_1^s)^\bullet, \dots, (t_n^s)^\bullet))]]$$

et s est bien dans le codomaine de $\psi^{\mathcal{L}}$.

III) Supposons $t = c[f^\alpha(t_1, \dots, t_n)] \rightarrow_e c[\langle x \rangle^\alpha u[t_1, \dots, t_n]] = t'$ par $\psi^{\mathcal{L}}$ -expansion et considérons un squelette s dans t' . Raisonnons par cas sur l'intersection de s avec c , $\langle x \rangle^\alpha u$ et les t_1, \dots, t_n . Notons $\alpha = [\Omega_n, \dots, [\Omega_1, p]]$ et remarquons que $\langle x \rangle^\alpha u = [\Omega_n, \dots, [\Omega_1, i_p(\langle x \rangle u^*)]]$.

- i) Si s est contenu dans c ou dans l'un des t_1, \dots, t_n , alors s est un squelette de t et pas hypothèse s est dans le codomaine de $\psi^{\mathcal{L}}$.
- ii) Si s est contenu dans $\langle x \rangle^\alpha u$ alors s a la bonne forme et appartient au codomaine de $\psi^{\mathcal{L}}$.
- iii) Si s intersecte à la fois $\langle x \rangle^\alpha u$ et au moins un de c ou des t_1, \dots, t_n , alors notons $s = c^s[\langle x \rangle^\alpha u[t_1^s, \dots, t_n^s]]$. L'ancêtre de s dans t est $s' = c^s[f^\alpha(t_1^s, \dots, t_n^s)]$, et par hypothèse s' est dans le codomaine de $\psi^{\mathcal{L}}$. En particulier

$$f^\alpha(t_1^s, \dots, t_n^s) = [\Omega_n, \dots, [\Omega_1, i_p(f((t_1^s)^\bullet, \dots, (t_n^s)^\bullet))]]$$

Donc

$$\langle x \rangle^\alpha u[t_1^s, \dots, t_n^s] = [\Omega_n, \dots, [\Omega_1, i_p(\langle x \rangle u^*[(t_1^s)^\bullet, \dots, (t_n^s)^\bullet])]]$$

et s est bien dans le codomaine de $\psi^{\mathcal{L}}$. □

Cet « axiome » de stabilité étant vérifié, nous héritons directement les constructions et résultats des sections 9.1 à 9.3. En particulier le système de *lifting* $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ stable pour $\Sigma^{\mathcal{L}}$ définit la réduction $\rightarrow_{l\mathcal{F}}$ de *méta-lifting* (définition 9.10) ainsi que les réductions intermédiaire \rightarrow_i et cible \rightarrow_f (définition 9.24), et les systèmes source et cible sont bisimilaires (théorème 9.35).

Convention 9.43. *Un CRS Σ et un système de lifting (\mathcal{F}, ψ) étant donnés, pour la fin de ce chapitre nous notons Σ^* le CRS regroupant les règles sources étiquetées, les règles intermédiaires étiquetées et les règles de méta-lifting étiquetées telles que définies par le système de lifting stable $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ proposé dans cette section.*

La réduction cible étiquetée définie par le système de *lifting* $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ utilise des noms de radicaux pour générer les étiquettes des membres droits, de même que la réduction source étiquetée que nous avons injectée dans le système. L'exemple 9.44 suivant montre ce comportement.

Exemple 9.44.

Considérons comme à l'exemple 9.18 la règle de réduction

$$\circ : \circ(\langle x \rangle Z_1(x), \langle y \rangle Z_2(y)) \rightarrow \langle z \rangle Z_1(Z_2(z))$$

Soient trois symboles h_x , h_y et h_z d'arités 0, 1 et 1 et munis des expansions

$$\begin{aligned} \psi(h_x) &= \langle x \rangle f(x, x) \\ \psi(h_y) &= \langle y \rangle \square \\ \psi(h_z) &= \langle z \rangle \square \end{aligned}$$

Considérons le terme étiqueté

$$t = \circ^\alpha(h_x^{[\Omega_0, p]}, h_y^\beta(a^\gamma))$$

Nous avons un pas de réduction cible étiquetée à partir de t , par la règle

$$\circ_f^l : \circ^\alpha(h_x^{[\Omega_0, p]}, h_y^\beta(Z_0)) \rightarrow h_z^{[\Omega, \epsilon]}(f^{[\Omega, 1]}(Z_0, Z_0))$$

où Ω est la séquence $\alpha[\Omega_0, p][\Omega_0, p \cdot 1]\beta$. En effet, cette règle \circ_f^l est obtenue par saturation de la règle source étiquetée

$$\circ_s^l : \circ^\alpha(\langle x \rangle^{[\Omega_0, p]} f^{[\Omega_0, p \cdot 1]}(x, x), \langle y \rangle^\beta(Z_0)) \rightarrow \langle z \rangle^{[\Omega, \epsilon]}(f^{[\Omega, 1]}(Z_0, Z_0))$$

dont le nom est hérité.

Ainsi, ce qui sert de nom à un pas de réduction cible étiquetée ρ_f est le nom Ω_s du pas de réduction source étiquetée ρ_s correspondant. En raison des contractions des motifs de règles, ce nom Ω_s contient a priori plus d'étiquettes que le nom Ω_f qui serait obtenu en appliquant directement l'étiquetage du chapitre 7 à la réduction cible non-étiquetée. Cependant, avec notre système $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ ces deux noms Ω_s et Ω_f ont mêmes contributeurs.

Enfin, si nous appliquons le même processus à l'étiquetage pleinement paresseux du chapitre 4, cette fois la réduction cible étiquetée serait directement égale à l'étiquetage pleinement paresseux de la réduction cible non-étiquetée.

9.4.2 Préservation des descendance

En combinant le lemme 9.31 de commutation entre réduction intermédiaire et *méta-lifting* avec l'utilisation des étiquettes comme témoins de la descendance, nous allons pouvoir montrer que les notions de descendance et de résiduation sont stables par *lifting*. Ceci permettra au chapitre 10 de déduire encore la stabilité de la nécessité.

Remarquons d'abord que les étiquettes, si elles sont bien utilisées, permettent de caractériser la relation de descendance le long d'une séquence de réduction CRS. Le lemme 9.45 suivant est une adaptation directe du lemme 7.37 qui avait été énoncé dans le cadre de l'étiquetage pleinement paresseux.

Lemme 9.45 (Descendance et étiquettes). *Soit t un terme étiqueté dont toutes les étiquettes sont initiales et différentes, et $t \rightarrow t'$ une séquence de réduction dans Σ^* . Alors une position $p' \in \text{pos}(t')$ descend d'une position $p \in \text{pos}(t)$ si et seulement si $\tau_{p'}(t') = \tau_p(t)$.*

Démonstration. De même qu'au lemme 7.37. □

Ceci nous permet d'abord d'étendre la fonction de *lifting* $\text{lft}(\cdot)$ aux positions des termes transformés.

Convention 9.46. *Considérons un terme étiqueté t dont les étiquettes sont toutes différentes. Par lemme 9.14 le terme étiqueté t a une unique forme normale $\text{lft}(t)$ pour le méta-lifting étiqueté. Donc par le lemme 9.45 toute séquence de méta-lifting de t à $\text{lft}(t)$ définit la même relation de descendance. Ainsi, pour toute position $p \in \text{pos}(t)$ nous noterons $\text{lft}(p)$ l'ensemble des descendants de p dans $\text{lft}(t)$, et cet ensemble est caractérisé en ce que pour toute séquence $\varrho_{\text{lft}} : t \rightarrow_{\text{lft}} \text{lft}(t)$,*

$$\text{lft}(p) = p/\varrho_{\text{lft}}$$

La notation s'étend aux ensembles de positions.

Nous pouvons maintenant énoncer la préservation par la fonction de *lifting* $\text{lft}(\cdot)$ de la descendance.

Lemme 9.47 (Préservation de la descendance). *Soient un terme source t , une position $p \in \text{pos}(t)$, et un pas de réduction source $\rho : t \rightarrow t'$. Alors*

$$\text{lft}(p/\rho) = \text{lft}(p)/\text{lft}(\rho)$$

Démonstration. Par théorème 9.35 de bisimulation nous avons le diagramme suivant :

$$\begin{array}{ccc} t & \xrightarrow{\varrho_{\text{lft}}} & \text{lft}(t) \\ | & & | \\ \rho & & \text{lft}(\rho) \\ \downarrow & & \downarrow \\ t' & \xrightarrow{\varrho'_{\text{lft}}} & \text{lft}(t') \end{array}$$

Par convention 9.46 et définition des descendants après une séquence nous avons

$$\text{lft}(p/\rho) = (p/\rho)/\varrho'_{\text{lft}} = p/(\rho\varrho'_{\text{lft}})$$

Or les séquences $\rho\varrho'_{\text{lft}}$ et $\varrho_{\text{lft}}\text{lft}(\rho)$ ont même cible, donc par lemme 9.45

$$p/(\rho\varrho'_{\text{lft}}) = p/(\varrho_{\text{lft}}\text{lft}(\rho))$$

De plus, par définition des descendants après une séquence et par convention 9.46 nous avons

$$p/(\rho_{\text{lft}} \text{lft}(\rho)) = (p/\rho_{\text{lft}})/\text{lft}(\rho) = \text{lft}(p)/\text{lft}(\rho)$$

Ainsi, en combinant ces trois étapes nous obtenons bien

$$\text{lft}(p/\rho) = \text{lft}(p)/\text{lft}(\rho)$$

□

De cette préservation de la descendance nous pouvons enfin déduire la préservation par *lifting* de la résiduation.

Lemme 9.48 (Préservation de la résiduation). *Soient trois pas de réduction sources $\rho : t \rightarrow t'$, $\rho_a : t \rightarrow u$, et $\rho_r : t' \rightarrow u'$. Alors ρ_r est un résidu de ρ_a après ρ si et seulement si $\text{lft}(\rho_r)$ est un résidu de $\text{lft}(\rho_a)$ après $\text{lft}(\rho)$.*

Démonstration. Par lemme 9.47 de préservation de la descendance, $\text{root}(\rho_r) \in \text{root}(\rho_a)/\text{redstep}$ si et seulement si $\text{root}(\text{lft}(\rho_r)) \in \text{root}(\text{lft}(\rho_a))/\text{lft}(\rho)$. Notons κ_a (resp. κ_r) la règle utilisée par le pas ρ_a (resp. ρ_r). Par convention 9.34, le pas $\text{lft}(\rho_a)$ (resp. $\text{lft}(\rho_r)$) utilise la règle $\text{lft}(\kappa_a)$ (resp. $\text{lft}(\kappa_r)$). En particulier $\kappa_a = \kappa_r$ implique $\text{lft}(\kappa_a) = \text{lft}(\kappa_r)$, et la réciproque est vraie également par convergence de l'expansion (lemme 9.16). Nous pouvons donc conclure. □

Ainsi la fonction de *lifting* préserve la descendance et la résiduation, et nous pouvons remarquer que ces propriétés prouvées pour le système avec étiquetage causal est héritée par le système non-étiqueté.

9.4.3 Préservation du partage

Nous pouvons vérifier que le CRS regroupant la réduction source étiquetée, la réduction intermédiaire étiquetée et le *méta-lifting* étiqueté forme un SLS. Remarquons que la propriété des squelettes uniformes (définition 8.44) qui était utilisée dans le cadre du λ -calcul (instanciation 8.53) n'apparaît plus explicitement ici. Elle est en effet couverte par la propriété de contractibilité appliquée au système de *lifting* $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$.

Pour cette vérification, nous aurons l'usage des lemmes 9.50 et 9.49 suivants qui assurent la commutation du *méta-lifting* étiqueté avec les opérations de manipulation des étiquettes.

Lemme 9.49 (*Lifting*/Étiquetage initial). *Considérons t et u deux termes sur la signature $\mathcal{S} \sqcup \mathcal{F}$ (non-étiquetée) et p une position. Alors $t \rightarrow_{\text{lft}} u$ (pour le méta-lifting issu de (\mathcal{F}, ψ)) si et seulement si $i_p(t) \rightarrow_{\text{lft}} i_p(u)$ (pour le méta-lifting issu de $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$).*

Démonstration. Par lemme 9.13, nous ramenons la preuve à la vérification pour \rightarrow_c et \rightarrow_c . Supposons

$$i_p(t) = c[\langle x \rangle^\alpha s[t_1, \dots, t_n]] \rightarrow_c c[f^\alpha(t_1, \dots, t_n)] = i_p(u)$$

Alors en particulier α est de la forme $p \cdot q$ et par définition

$$\langle x \rangle^\alpha s = \psi^{\mathcal{L}}(f^\alpha) = i_{p \cdot q}(\psi(f))$$

En particulier $\langle x \rangle s^*$ est dans le codomaine de ψ , et nous avons $t \rightarrow_c u$. De même si $t \rightarrow_c u$ alors $i_p(t) \rightarrow_c i_p(u)$, et $t \rightarrow_e u$ si et seulement si $i_p(t) \rightarrow_e i_p(u)$. \square

Lemme 9.50 (*Lifting/Réétiquetage uniforme*). Soient t et u deux termes et Ω une séquence d'étiquettes. Alors $t \rightarrow_{ift} u$ si et seulement si $[\Omega, t] \rightarrow_{ift} [\Omega, u]$.

Démonstration. Par lemme 9.13, nous ramenons la preuve à la vérification pour \rightarrow_c et \rightarrow_e . Supposons

$$t = c[\langle x \rangle^\alpha s[t_1, \dots, t_n]] \rightarrow_c c[f^\alpha(t_1, \dots, t_n)] = u$$

avec $\alpha = [\Omega_n, \dots [\Omega_1, p]]$. Alors par définition

$$\langle x \rangle^\alpha s = \psi^{\mathcal{L}}(f^\alpha) = [\Omega_n, \dots [\Omega_1, i_p(\psi(f))]]$$

En particulier, $[\Omega, \langle x \rangle^\alpha s]$ est toujours dans le codomaine de ψ , et plus précisément

$$[\Omega, \langle x \rangle^\alpha s] = \psi^{\mathcal{L}}(f^{[\Omega, \alpha]}) = [\Omega, [\Omega_n, \dots [\Omega_1, i_p(\psi(f))]]]$$

Alors $[\Omega, t] \rightarrow_c [\Omega, u]$. De même si $[\Omega, t] \rightarrow_c [\Omega, u]$ alors $t \rightarrow_c u$, et $t \rightarrow_e u$ si et seulement si $[\Omega, t] \rightarrow_e [\Omega, u]$. \square

Instanciation 9.51. Considérons le triplet $(\mathcal{S}^*, \mathcal{T}^*, \mathcal{R}^*)$ formé de :

- la signature $\mathcal{S}^* = (\mathcal{S} \sqcup \mathcal{F} \sqcup \{\langle x \rangle \mid x \in \mathcal{X}\})^{\mathcal{L}}$,
- l'ensemble \mathcal{T}^* des termes vérifiant la propriété de contractibilité pour $\psi^{\mathcal{L}}$,
et
- l'ensemble $\mathcal{R}^* = \mathcal{R}_s \cup \mathcal{R}_i \cup \mathcal{R}_{ift}$ où \mathcal{R}_s est l'ensemble des pas de réduction source dont la source vérifie la propriété de contractibilité pour $\psi^{\mathcal{L}}$, \mathcal{R}_i l'ensemble des pas de réduction intermédiaire dont la source vérifie la propriété de contractibilité pour $\psi^{\mathcal{L}}$, et \mathcal{R}_{ift} l'ensemble des pas de méta-lifting dont la source vérifie la propriété de contractibilité pour $\psi^{\mathcal{L}}$.

Pour tout pas de réduction $\rho \in \mathcal{R}^*$ notons $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et définissons :

- $\text{dex}(\rho) = l^\sigma$
- $\text{duct}(\rho) = r^\sigma$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \mathcal{P}_p(r, \sigma)$

Lemme 9.52. Le système $\Sigma^* = (\mathcal{S}^*, \mathcal{T}^*, \mathcal{R}^*, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$ est un ATRS marqué.

Démonstration.

Source & Cible Par lemme 9.42 la propriété de contractibilité est préservée par les pas de réduction \rightarrow , \rightarrow_c , et \rightarrow_e . Par lemme 9.13 et remarque 9.23 cette propriété est donc encore préservée par les pas de réduction \rightarrow_i et \rightarrow_{ift} .

Résidus Par stabilité des positions gelées.

Zone d'effet De même qu'à la section 4.3, tout sous-terme du réduit hors de la zone d'effet est contenu dans le codomaine de σ et est donc déjà présent dans le radical. \square

Lemme 9.53. *Le triplet $(\mathcal{L}, \leq, \Sigma^*)$ est un SLS.*

Démonstration.

Étiquette racine Les symboles sont tous étiquetés.

Progression Il faut distinguer deux cas selon le type de pas de réduction considéré.

- Dans le cas d'un pas de réduction $\rho \in \mathcal{R}_s \cup \mathcal{R}_i$, toute étiquette de la zone d'effet a la forme $[\Omega, \alpha]$ avec $\Omega = \|\iota\| = \tau(\iota)\Omega'$, et en particulier $\tau(\iota) < [\Omega, \alpha]$.
- Dans le cas d'un pas de réduction $\rho \in \mathcal{R}_{ift}$, la zone d'effet est un singleton, dont l'unique étiquette est $\tau(\iota)$ par la remarque 9.41.

Héritage De même qu'à la section 4.3.

Partage Séparons trois cas en fonction du type de pas de réduction $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ considéré.

- Cas $\rho \in \mathcal{R}_s$. Alors le membre droit r est de la forme $[\Omega, i(r_0)]$. Donc toutes les étiquettes de la zone d'effet sont différentes.
- Cas $\rho \in \mathcal{R}_i$. Alors le membre droit r est de la forme $\text{1ft}([\Omega, i(r_0)])$, d'où $r = [\Omega, \text{1ft}(i(r_0))]$ par lemme 9.50 et $r = [\Omega, i(\text{1ft}(r_0))]$ par lemme 9.49. Donc toutes les étiquettes de la zone d'effet sont différentes.
- Cas $\rho \in \mathcal{R}_{ift}$. Alors la zone d'effet est un singleton. \square

Nous avons donc montré d'une part que le *méta-lifting* définissait une bisimulation entre la réduction de notre système étiqueté source et la réduction étiquetée cible associée (théorème 9.35 de bisimulation, qui est applicable par lemme 9.42 de stabilité de notre système de *lifting* étiqueté $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$) et d'autre part que le *méta-lifting* parallèle étiqueté préservait la propriété de partage (théorème 1.19 de partage, qui est applicable par lemme 9.53 de vérification des axiomes des *SLS*). Pour en déduire que le *méta-lifting* induit bien une bisimulation entre les réductions parallèles étiquetées source et cible, il reste à vérifier que les mêmes radicaux sont partagés avant et après *méta-lifting*. Cette propriété correspond aux résultats de « partage équivalent », dont sont par exemple le théorème 7.29 d'équivalence entre les partages spécifiés par l'étiquetage causal et l'étiquetage pleinement paresseux d'un *CRS* et le lemme 7.35 de préservation du partage par expansion/compression d'un système.

Ici, nous ne prendrons pas l'approche des termes miroirs utilisée dans les preuves des théorèmes 3.26, 4.52, et 7.29, qui consistait à considérer des termes issus d'une même source par des séquences de réduction équivalentes (« miroirs »). Comme au lemme 7.35, nous profiterons au contraire du fait que nous avons une transformation directe entre les deux systèmes étudiés (le système source et le système cible), et que cette transformation est décrite par un système

de réécriture, pour simplement vérifier la stabilité des étiquettes par descendance le long des pas de *méta-lifting*.

Nous pouvons alors déduire que le *méta-lifting* préserve le partage.

Lemme 9.54 (Préservation du partage). *Soient t et u deux termes sur $(\mathcal{S} \sqcup \mathcal{F})^{\mathcal{L}}$ tels que $t \Rightarrow_{\text{ift}} u$. Soit p_u (resp. q_u) une position de u admettant un ancêtre p_t (resp. q_t) dans t . Alors $\tau_{p_t}(t) = \tau_{q_t}(t)$ si et seulement si $\tau_{p_u}(u) = \tau_{q_u}(u)$.*

Démonstration. Par lemme 1.8 de réduction de la simulation parallèle il existe une séquence de réduction \rightarrow_{ift} de t à u , et nous pouvons appliquer le lemme 9.45 à cette séquence. Donc si $\tau_{p_t}(t) = \tau_{q_t}(t)$, alors par lemme 9.45 $\tau_{p_u}(u) = \tau_{p_t}(t) = \tau_{q_t}(t) = \tau_{q_u}(u)$. \square

Théorème 9.55 (Bisimulation). *La fonction de lifting $\text{lift}(\cdot)$ est une bisimulation entre la réduction parallèle étiquetée du système source et la réduction parallèle étiquetée du système cible.*

Finalement, la réduction partagée de notre *CRS* étiqueté source (l'étiquetage causal d'un *CRS* selon le chapitre 7) est bisimilaire à la réduction partagée du *CRS* étiqueté cible, qui est en réalité un *TRS*. Or, depuis la section 7.3 nous savons de plus que l'étiquetage causal du chapitre 7 et l'étiquetage pleinement paresseux du chapitre 4 définissent la même réduction parallèle étiquetée. Ainsi, la réduction pleinement paresseuse d'un *CRS* telle que définie au chapitre 4 est bisimilaire à la réduction partagée d'un certain système du premier ordre.

Bilan du chapitre

Dans ce chapitre nous avons généralisé au cadre des *CRS* la vision du λ -lifting en tant que transformation entre systèmes de réécriture introduite au chapitre précédent.

Nous avons obtenu une transformation qui à tout *CRS* faible associe un système de réécriture du premier ordre qui lui est bisimilaire. Cette transformation préserve de plus l'éventuelle orthogonalité des systèmes auxquels elle est appliquée, ainsi que les notions de descendance et de résiduation.

Comme au chapitre précédent cette transformation a été elle-même décrite comme un système de réécriture convergent. Avec cette vision "à petits pas" de l'opération globale nous avons gagné une preuve extrêmement concise de la bisimilarité entre la source et la cible.

En outre nous avons pu bénéficier dans ce chapitre de la nature extensible des *CRS*, qui a permis d'une part d'exprimer directement le lifting comme un ensemble de règles *CRS* et d'autre part de voir le lifting étiqueté comme une simple instance du lifting général.

Nous avons enfin pu prouver que cette opération de lifting définissait une bisimulation entre la réduction pleinement paresseuse telle que définie au chapitre 4 et une certaine réduction partagée des systèmes de réécriture du premier ordre, dont nous allons voir dans la partie suivante qu'elle correspond,

dans le cas d'un système orthogonal, à la réduction optimale. Nous pourrions y énoncer, enfin, le caractère optimal de la réduction pleinement paresseuse.

CINQUIÈME PARTIE

OPTIMALITÉS

Introduction

Cette ultime partie conclut la thèse en considérant les deux faces des résultats d'optimalité vers lesquels se sont acheminées les parties précédentes.

Le chapitre 10 d'abord, qui représente la face positive, énonce les résultats d'opimalité que vérifie sur les systèmes orthogonaux la réduction pleinement paresseuse telle que définie à la partie II, à savoir l'optimalité dans l'espace des réductions faibles sans partage et l'optimalité dans l'espace des réductions de familles faibles.

Le chapitre 11 enfin, qui représente la *vraie* face positive, s'affranchit partiellement des restrictions liées à la réduction faible et à la causalité qui sont les limites de l'*optimalité faible* énoncée au chapitre 10 et montre comment les outils présentés dans cette thèse permettent d'aller *au-delà*.

CHAPITRE 10

Optimalité faible

Sommaire

10.1 Optimalité faible sans partage	310
10.2 Non-calculabilité des stratégies optimales	312
10.3 Partage et effectivité	319
10.4 Un cadre axiomatique pour les familles	321
10.5 Optimalité faible	325

Dans ce chapitre nous récoltons les fruits de ce que nous avons semé dans les parties précédentes. À l'aide de la traduction des *CRS* faibles au premier ordre donnée par la partie IV, de la notion de partage pleinement paresseux définie à la partie II, et des propriétés causales des systèmes orthogonaux établies à la partie III, nous allons obtenir plusieurs caractérisations des stratégies de réduction optimales pour les *CRS* faibles orthogonaux.

Ces résultats d'optimalité ne demanderont que peu de travail neuf. En effet, le travail préparatoire effectué dans les parties précédentes permet de se ramener quasi-immédiatement à différents résultats de la littérature concernant soit les systèmes de réécriture du premier ordre orthogonaux soit des systèmes axiomatiques. Ainsi, quatre des cinq sections de ce chapitre consistent essentiellement en :

- énoncer des résultats de la littérature, et/ou
- vérifier la bonne application de ces résultats à notre cadre.

La section 10.1 d'abord décrit des stratégies de réduction donnant dans les *CRS* faibles orthogonaux des séquences de réduction sans partage de longueur minimale. Ces stratégies et la preuve de leur optimalité sont déduites d'un résultat similaire de Zurab Khasidashvili sur les systèmes de réécriture du premier ordre orthogonaux [Kha93] au moyen de la correspondance entre *CRS* faibles et systèmes du premier ordre donnée par le *lifting* généralisé au chapitre 9.

La section 10.2 tempère ce premier résultat en démontrant qu'en λ -calcul faible, les stratégies de réduction sans partage *optimales* (c'est-à-dire qui génèrent dans tous les cas des séquences de réduction de longueur minimale) ne peuvent être calculables. Ce fait avait déjà été établi dans le cadre usuel du

λ -calcul non-faible par Barendregt, Bergstra, Klop, et Volken [BBKV76], et nous prendrons leur preuve comme point de départ. Cependant, l'adaptation est loin d'être immédiate. Notre nouvelle preuve sera en revanche l'occasion d'avancer une utilisation originale de la notion de squelette et des idées du λ -*lifting*, comme base de la décomposition d'un λ -terme en un ensemble de composantes normales.

La section 10.3 montre ensuite que les réductions pleinement paresseuses sont équivalentes aux réductions optimales sans partage décrites à la section 10.1. Ce point a deux conséquences. Nous y constatons d'abord que le partage pleinement paresseux ne permet pas d'obtenir des séquences de réduction strictement plus courtes que les séquences optimales sans partage. Mais en contrepartie, le partage pleinement paresseux nous donne de nouveaux moyens d'atteindre cette borne inférieure commune, et permet dans de nombreux cas d'avoir pour cela des stratégies calculables. Ici finalement, le partage est un moyen de gagner non sur la borne des longueurs des séquences de réduction, mais sur leur atteignabilité par des stratégies calculables.

Les sections 10.4 et 10.5 enfin montrent que les stratégies pleinement paresseuses réalisent l'optimalité à la Lévy pour les CRS faibles orthogonaux, c'est-à-dire qu'elles sont optimales parmi les stratégies avec partage respectant les familles de pas de réduction définies par la structure causale des CRS faibles orthogonaux. Cette conclusion est cette fois déduite d'un résultat axiomatique de Glauert et Khasidashvili sur la réduction partagée dans des ARS munis d'une notion de résiduation [GK96], résultat qui est rappelé à la section 10.4. Cette axiomatisation non triviale du partage est utilisée à la section 10.5, grâce en particulier aux étiquettes causales introduites au chapitre 7 et à leurs propriétés de correction et de complétude.

Nous obtenons à l'issue de ce chapitre trois notions concordantes d'optimalité pour les CRS orthogonaux en réduction faible (optimalité sans partage, réduction pleinement paresseuse, familles de Lévy). Elles définissent et atteignent une même borne sur la longueur des réduction faibles, borne que nous appelons *optimalité faible*.

10.1 Optimalité faible sans partage

Cette section s'applique à tous les CRS faibles orthogonaux, et définit une condition suffisante pour qu'une séquence de réduction faible soit optimale dans l'espace des réductions faibles sans partage. Nous nous basons pour cela sur un résultat de Khasidashvili [Kha93], qui assure que dans tout système de réécriture du premier ordre orthogonal les séquences de réduction *innermost needed* (réduction de *radicaux nécessaires les plus profonds*) sont toujours les plus courtes. Ces résultats sont transférés au cadre des CRS faibles orthogonaux grâce au *lifting* défini au chapitre 9 et aux résultats associés de bisimulation (théorème 9.35) et de préservation de la résiduation (lemme 9.48).

Rappelons d'abord la notion de réduction nécessaire, qui s'exprime dans un cadre abstrait et est donc indépendante du format de réécriture choisi.

Définition 10.1 (Pas de réduction nécessaire). *Considérons un ARS muni d'une opération de résiduation et un objet t normalisable. Un pas de réduction ρ de source t est **nécessaire** si toute séquence de réduction menant de t à une forme normale contient au moins un résidu de ρ .*

Le résultat de Khasidashvili s'énonce alors ainsi :

Théorème 10.2 ([Kha93]). *Considérons t un terme normalisable dans un TRS orthogonal, et ρ une séquence de réduction normalisante de source t dont chaque pas réduit un radical nécessaire le plus profond. Alors aucune réduction normalisante de source t n'est strictement plus courte que ρ .*

Considérant un CRS orthogonal Σ et un système de *lifting* stable associé (définition 9.9), le *lifting* présenté au chapitre 9 nous fournit un TRS orthogonal $\text{lft}(\Sigma)$ équivalent à Σ , dans lequel le théorème 9.35 de bisimulation permet de lier toute séquence de réduction d'un système à une séquence de réduction de même longueur de l'autre système.

Ainsi, le théorème 10.2 de Khasidashvili nous assure que les séquences nécessaires profondes de $\text{lft}(\Sigma)$ décrivent par l'intermédiaire du *lifting* des séquences de réduction de Σ de longueur minimale. Pour assurer dans les CRS un résultat similaire au théorème 10.2 il faut cependant encore pouvoir affirmer que les séquences de réduction ainsi construites dans Σ sont bien des séquences nécessaires profondes. Ce point nécessitera l'utilisation du lemme 10.3 de préservation par *lifting* de la notion de nécessité.

Lemme 10.3 (Stabilité de la nécessité). *Considérons $\Sigma = (\mathcal{S}, \mathcal{R})$ un CRS faible orthogonal et (\mathcal{F}, ψ) un système de *lifting* stable pour Σ . Soit t un terme normalisable de Σ . Alors un pas de réduction ρ de source t est nécessaire si et seulement si son image $\text{lft}(\rho)$ dans le système cible $\text{lft}(\Sigma)$ est nécessaire.*

Démonstration. Considérons un pas de réduction ρ de source t qui est nécessaire. Soit $\rho^f = \rho_1^f \dots \rho_n^f$ une séquence de réduction dans $\text{lft}(\Sigma)$ qui est normalisante et de source $\text{lft}(t)$. Par théorème 9.35 de bisimulation il existe une séquence de réduction $\rho = \rho_1 \dots \rho_n$ dans Σ qui est normalisante et de source t . Par hypothèse ρ est nécessaire, il existe donc un ρ_i qui est un résidu de ρ . Alors par lemme 9.48 de préservation de la résiduation, le pas de réduction ρ_i^f est un résidu du pas de réduction $\text{lft}(\rho)$. Ainsi $\text{lft}(\rho)$ est nécessaire.

Nous prouvons de même que si $\text{lft}(\rho)$ est nécessaire, alors ρ l'est aussi. \square

Nous avons maintenant toutes les cartes en main pour démontrer l'optimalité des réductions nécessaires profondes dans les CRS faibles orthogonaux.

Théorème 10.4 (Optimalité sans partage). *Considérons t un terme normalisable dans un CRS orthogonal faible Σ , et ρ une séquence de réduction normalisante de source t dont chaque pas réduit un radical nécessaire le plus profond. Alors aucune réduction normalisante de source t n'est strictement plus courte que ρ .*

Démonstration. Supposons qu'existe une séquence de réduction normalisante ρ' de source t qui soit strictement plus courte que ρ . Nous allons montrer que ceci contredit le théorème 10.2 d'optimalité dans les systèmes du premier ordre orthogonaux.

Soit (\mathcal{F}, ψ) un système de lifting stable pour Σ . Notons Σ_f le système du premier ordre cible associé à Σ , et notons ρ_f (resp. ρ'_f) l'image de ρ (resp. ρ') dans Σ_f . Par théorème 9.35 de bisimulation les séquences ρ_f et ρ'_f sont deux séquences de réduction normalisantes co-initiales. De plus, et par théorème 9.35 toujours, ρ_f (resp. ρ'_f) a la même longueur que ρ (resp. ρ'). En particulier ρ'_f est strictement plus courte que ρ_f .

Par théorème 9.37 de préservation de l'orthogonalité le système Σ_f est orthogonal, et par lemme 10.3 de stabilité de la notion de nécessité la séquence ρ_f est une séquence de réduction normalisante dont chaque pas réduit un radical nécessaire le plus profond. Le théorème 10.2 s'applique donc et assure qu'il n'existe dans Σ_f aucune séquence de réduction normalisante co-initiale à ρ_f strictement plus courte que ρ_f : les séquences ρ'_f et ρ' ne peuvent exister. \square

Avec ce théorème 10.4 d'optimalité sans partage nous obtenons une condition suffisante pour qu'une séquence de réduction normalisante dans un CRS faible orthogonal soit de longueur minimale. Nous allons cependant voir à la section suivante que malgré cette condition, dans de nombreux systèmes il est impossible de définir une stratégie calculable fournissant de telles séquences normalisantes de longueur minimale.

10.2 De la non-calculabilité des stratégies de réduction optimales en λ -calcul faible

Barendregt, Bergstra, Klop et Volken [BBKV76, Bar84] ont montré qu'une stratégie de réduction (sans partage) pour le λ -calcul générant pour tout λ -terme normalisable une séquence de réduction normalisante la plus courte ne pouvait être calculable. L'objectif de cette section est la démonstration du théorème 10.6 exprimant que cette non-calculabilité vaut encore pour le λ -calcul restreint à la réduction faible, et donc par ricochet vaut encore pour de nombreux CRS faibles.

Nous reprendrons la structure de la preuve de Barendregt et al. telle que relatée par Barendregt [Bar84], qui est en grande partie compatible avec la réduction faible. Nous verrons cependant qu'une des étapes de leur construction n'est plus valide en réduction faible, et l'essentiel de cette section se concentrera sur la résolution de cette étape. Cette nouvelle résolution fera jouer un rôle central à la notion de squelette d'une abstraction, et utilisera également certaines idées sous-tendant le λ -lifting qui ont été égrenées au chapitre 8.

Précisons deux points de vocabulaire :

Définition 10.5 (Stratégies optimales). Une **stratégie** est une fonction F qui à tout λ -terme t hors forme normale associe un λ -terme $F(t) = t'$ tel que $t \rightarrow t'$. Une stratégie est **optimale** si les séquences de réduction qu'elle génère sont toujours de longueur minimale.

Le théorème que nous prouvons est le suivant :

Théorème 10.6. *Il n'existe pas de stratégie optimale et calculable pour le λ -calcul faible.*

La preuve de Barendregt et al. [BBKV76, Bar84] procède en exhibant deux ensembles récursivement énumérables de λ -termes \mathcal{A} et \mathcal{B} qui ne sont pas récursivement séparables¹, et en montrant comment séparer ces deux ensembles à l'aide d'une hypothétique stratégie de réduction optimale calculable (aussi appelée *récursive*).

Les deux ensembles \mathcal{A} et \mathcal{B} sont construits à l'aide un théorème dû à Scott [Sco68] qui assure que deux ensembles de λ -termes non vides et clos par β -conversion sont nécessairement récursivement inséparables. Barendregt et al. retiennent donc comme forme pour \mathcal{A} (resp. \mathcal{B}) « l'ensemble des λ -termes admettant pour forme normale un certain t_a » (resp. t_b), où les formes normales t_a et t_b sont bien choisies. Enfin, ils construisent une énumération t_0, t_1, t_2, \dots de l'ensemble des λ -termes, et une suite u_0, u_1, u_2, \dots de λ -termes telle que :

- Chaque u_k contient exactement deux β -radicaux r_k^a et r_k^b .
- Dans u_k , toute stratégie optimale F contracte r_k^a si $t_k \twoheadrightarrow t_a$, et contracte r_k^b si $t_k \twoheadrightarrow t_b$ (et rien n'est spécifié si t_k ne se réduit ni à t_a ni à t_b).

Ainsi le comportement d'une stratégie optimale F sur les u_k permet de séparer les t_k appartenant à \mathcal{A} des t_k appartenant à \mathcal{B} , et une telle fonction ne peut donc pas être calculable.

Pour reproduire cette preuve dans un cadre de réduction faible, il faut prêter attention à trois points :

1. le résultat général d'inséparabilité,
2. l'énumération t_0, t_1, t_2, \dots de l'ensemble des λ -termes, et
3. l'intégration de l'énumération dans une suite u_0, u_1, u_2, \dots de λ -termes sur lesquels le comportement d'une stratégie optimale est discriminant.

Le premier point ne pose pas de difficultés (théorème 10.8). Le deuxième en revanche doit être complètement revu en réduction faible, révision qui impose alors d'introduire de nouvelles idées également pour le troisième point.

Concentrons nous donc d'abord sur le deuxième point : l'énumération de tous les λ -termes. Nous allons d'abord voir en quoi la technique de Barendregt et al. ne s'applique pas en réduction faible, puis proposer une alternative. Leur technique est basée sur la traduction du λ -calcul en logique combinatoire [CF74] et tire parti de la remarque suivante :

- Prenons un λ -terme t .
Par exemple : $t = \lambda x.xx$
- Notons $(t)_{CL}$ sa traduction usuelle sur la base des combinateurs S, K et I .
Pour notre exemple : $(t)_{CL} = SII$

1. Rappelons que deux ensembles \mathcal{A} et \mathcal{B} sont *récursivement séparables* s'il existe un ensemble \mathcal{C} récursif (syn. calculable) tel que $\mathcal{A} \subseteq \mathcal{C}$ et $\mathcal{B} \cap \mathcal{C} = \emptyset$.

- Notons $((t)_{CL})_\lambda$ le terme $(t)_{CL}$ dans lequel chaque S est remplacé par $\lambda xyz.xz(yz)$, chaque K est remplacé par $\lambda xy.x$, et chaque I est remplacé par $\lambda x.x$.

$$\text{En l'occurrence : } ((t)_{CL})_\lambda = (\lambda xyz.xz(yz))(\lambda x.x)(\lambda x.x)$$

- Alors $((t)_{CL})_\lambda \rightarrow t$.

$$\begin{aligned} \text{Effectivement : } ((t)_{CL})_\lambda &\rightarrow (\lambda yz.(\lambda x.x)z(yz))(\lambda x.x) \\ &\rightarrow \lambda z.(\lambda x.x)z((\lambda x.x)z) \\ &\rightarrow \lambda z.z((\lambda x.x)z) \\ &\rightarrow \lambda z.zz \end{aligned}$$

Les λ -termes peuvent alors être énumérés indirectement à partir de l'énumération des combinaisons de combinateurs. Problème pour nous : retrouver t à partir de $((t)_{CL})_\lambda$ demande des réductions à des positions gelées (les deux derniers pas dans l'exemple). Cette dernière remarque n'est donc plus valide en réduction faible.

L'objectif de l'énumération indirecte de Barendregt et al. à partir des combinateurs était de contourner la difficulté suivante : énumérer les λ -termes clos en suivant leur grammaire pose problème, puisque les λ -abstractions sont susceptibles d'introduire une quantité arbitraire de nouvelles variables. Nous allons en revanche nous concentrer sur un sous-ensemble du λ -calcul sur lequel cette énumération directe est possible. Nous allons considérer dans cette section des λ -termes dont toutes les variables libres et liées appartiennent à un ensemble énumérable fixé.

Définition 10.7 (Lambda-termes restreints). *Pour tout entier $n \in \mathbb{N}$, soit x_n une variable. Notons $\mathcal{X}_r = \{x_n \mid n \in \mathbb{N}\}$ leur ensemble. Les **λ -termes restreints** sont les λ -termes dont toutes les variables libres et liées appartiennent à \mathcal{X}_r . Leur ensemble est noté Λ_r et est donné par la grammaire*

$$t ::= x_n \mid \lambda x_n.t \mid tt$$

Le changement a l'air minime, et effectivement ce λ -calcul a les mêmes propriétés que le λ -calcul usuel, notamment pour ce qui concerne les ensembles récursivement inséparables.

Théorème 10.8 (Directement adapté de Scott [Sco68]). *Soient $\mathcal{A}, \mathcal{B} \subseteq \Lambda_r$ deux ensembles non-vides de λ -termes restreints qui sont clos par β -conversion faible. Alors \mathcal{A} et \mathcal{B} ne sont pas récursivement séparables.*

Esquisse de démonstration. Identique au compte-rendu donné par Barendregt [Bar84, Chap. 6] de cette preuve. Attention cependant, cette preuve est à tiroirs, puisqu'elle fait appel à des résultats classiques de points fixes et de définissabilité dans le λ -calcul. C'est donc de la majorité du chapitre [Bar84, Chap. 6] qu'il faut vérifier la compatibilité avec la réduction faible, et nous avons besoin au passage de la propriété de Church-Rosser pour le λ -calcul faible (prouvée par Çağman et Hindley [cH98]). \square

La différence est que nous pouvons maintenant facilement construire une bijection récursive entre l'ensemble des entiers naturels \mathbb{N} et cet ensemble restreint Λ_r . Fixons d'abord un outil de base :

Convention 10.9. Fixons une fonction paire bijective et récursive

$$\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

munie de deux fonctions de projection récursives $\pi_1(\cdot)$ et $\pi_2(\cdot)$ (ces fonctions sont nombreuses, nous pouvons considérer par exemple celle qui construit la paire en intercalant les bits des deux arguments).

Ces fonctions vérifient les conditions suivantes

$$\begin{aligned} \forall k \in \mathbb{N} \quad \langle \pi_1(k), \pi_2(k) \rangle &= k \\ \forall k_1, k_2 \in \mathbb{N} \quad \pi_1(\langle k_1, k_2 \rangle) &= k_1 \\ \forall k_1, k_2 \in \mathbb{N} \quad \pi_2(\langle k_1, k_2 \rangle) &= k_2 \end{aligned}$$

Grâce à la fonction paire $\langle \cdot, \cdot \rangle$ et ses projections $\pi_1(\cdot)$ et $\pi_2(\cdot)$, nous pouvons définir une bijection récursive entre \mathbb{N} et Λ_r .

Définition 10.10 (Codage, décodage). Pour tout λ -terme restreint t , les équations inductives suivantes définissent un entier $\text{enc}(t)$ appelé **codage** de t .

$$\begin{aligned} \text{enc}(x_n) &= 3 * n \\ \text{enc}(\lambda x_n. t) &= 1 + 3 * \langle n, \text{enc}(t) \rangle \\ \text{enc}(t_1 t_2) &= 2 + 3 * \langle \text{enc}(t_1), \text{enc}(t_2) \rangle \end{aligned}$$

Réciproquement, à tout entier k les équations inductives suivantes définissent un λ -terme restreint $\text{dec}(k)$ appelé **décodage** de k .

$$\begin{aligned} \text{dec}(3k) &= x_k \\ \text{dec}(1 + 3k) &= \lambda x_{\pi_1(k)}. \text{dec}(\pi_2(k)) \\ \text{dec}(2 + 3k) &= \text{dec}(\pi_1(k)) \text{dec}(\pi_2(k)) \end{aligned}$$

Lemme 10.11 (Codage et décodage). Pour tout λ -terme restreint $t \in \Lambda_r$ nous avons $\text{dec}(\text{enc}(t)) = t$. Pour tout entier $k \in \mathbb{N}$ nous avons $\text{enc}(\text{dec}(k)) = k$.

Démonstration. Par induction sur t (resp. par récurrence forte sur k). \square

Ainsi, en énumérant les entiers naturels, via notre fonction récursive de décodage nous énumérons également les λ -termes restreints.

Nous pouvons donc maintenant passer au troisième point : l'intégration de ces λ -termes $t_0, t_1, t_2, \dots = \text{dec}(0), \text{dec}(1), \text{dec}(2), \dots$ que nous énumérons dans une suite u_0, u_1, u_2, \dots de λ -termes sur lesquels le comportement d'une stratégie optimale F est discriminant. Pour cela nous avons demandé que, indépendamment du t_k associé, chaque u_k contienne exactement deux radicaux, entre lesquels la stratégie F doit choisir. Barendregt et al. réussissent ceci en « cachant » le λ -terme t_k dans deux sous-termes e et n_k de u_k qui sont chacun en forme normale. Le terme u_k est en l'occurrence de la forme :

$$u_k = (\lambda x. x e x)(\lambda y. y n_k (II))$$

L'idée est d'avoir une forme normale e qui renferme une fonction d'énumération, et une forme normale n_k qui représente l'entier k . À un certain point, les

réductions de u_k vont faire apparaître une application en_k , ce qui permettra avec un peu plus de réduction de générer le terme t_k correspondant (ou au moins des termes qui sont β -convertibles à t_k et ont donc même forme normale par propriété de Church-Rosser).

Nous allons de même utiliser une décomposition de nos termes énumérés $t_k = dec(k)$ en plusieurs formes normales. Et c'est à ce point que vont intervenir les idées du λ -lifting. En λ -calcul faible, les termes contiennent un certain nombre de composantes « en forme normale » : leurs squelettes. Nous allons donc décomposer un λ -terme en un assemblage de formes normales en isolant ses squelettes, de la même manière que le fait le λ -lifting, mais sans faire appel à la définition au niveau « méta » d'un expanseur.

Dans une première étape nous réorganiserons le terme à décomposer en une combinaison applicative de variables libres et de squelettes d'arité nulle, c'est-à-dire des squelettes qui sont des sous-termes entiers et non des contextes (décomposition en squelettes, définition 10.12). Cette étape de transformation n'était pas explicite dans la formalisation du λ -lifting utilisée dans ce chapitre. Elle correspond bien cependant à l'une des deux actions qu'effectue le λ -lifting, caractérisée par Danvy et Schultz par l'expression *parameter lifting* [DS00]. Ensuite, nous extrairons les squelettes par β -expansion de manière à obtenir un terme de la forme $s_0 s_1 \dots s_n$ où les s_i sont tous des formes normales (abstraction des squelettes, définition 10.16). En quelque sorte, le terme s_0 correspond au résultat de la transformation de λ -lifting, tandis que les s_i sont les définitions (les expansions !) des symboles de supercombinateurs utilisés dans le programme principal s_0 . Ceci correspond à la deuxième et dernière action qu'effectue le λ -lifting, caractérisée par l'expression *block floating* [DS00].

Définition 10.12 (Décomposition en squelettes). *Pour tout λ -terme t , les équations inductives suivantes définissent un λ -terme $\llbracket t \rrbracket$ appelé **décomposition en squelettes** de t .*

$$\begin{aligned} \llbracket x \rrbracket &= x \\ \llbracket t_1 t_2 \rrbracket &= \llbracket t_1 \rrbracket \llbracket t_2 \rrbracket \\ \llbracket \lambda x. t \rrbracket &= (\lambda y_1 \dots y_n x. s[y_1, \dots, y_n]) \llbracket t_1 \rrbracket \dots \llbracket t_n \rrbracket \\ &\quad s = \langle\langle t \rangle\rangle^{(x)} \\ &\quad t = s[t_1, \dots, t_n] \\ &\quad y_1, \dots, y_n \text{ variables fraîches} \end{aligned}$$

Lemme 10.13 (Décomposition en squelettes). *Pour tout λ -terme t il existe une séquence de β -réduction faible*

$$\varrho : \llbracket t \rrbracket \twoheadrightarrow t$$

Démonstration. Par induction sur t .

- Le cas x est immédiat.
- Cas $t_1 t_2$. Par hypothèse d'induction $\llbracket t_1 \rrbracket \twoheadrightarrow t_1$ et $\llbracket t_2 \rrbracket \twoheadrightarrow t_2$. Donc

$$\llbracket t_1 t_2 \rrbracket = \llbracket t_1 \rrbracket \llbracket t_2 \rrbracket \twoheadrightarrow t_1 \llbracket t_2 \rrbracket \twoheadrightarrow t_1 t_2$$

- Cas $\lambda x.t$. Notons $s = \langle\langle t \rangle\rangle^{\{x\}}$ et $t = s[t_1, \dots, t_n]$. Alors il existe n variables y_1, \dots, y_n telles que $\llbracket \lambda x.t \rrbracket = (\lambda y_1 \dots y_n x. s[y_1, \dots, y_n])[t_1] \dots [t_n]$. Or par hypothèse de récurrence, pour tout $i \in \{1, \dots, n\}$ $\llbracket t_i \rrbracket \rightarrow t_i$. Donc

$$\begin{aligned} \llbracket \lambda x.t \rrbracket &\rightarrow (\lambda y_1 \dots y_n x. s[y_1, \dots, y_n])t_1 \dots t_n \\ &\rightarrow^n \lambda x. s[t_1, \dots, t_n] \end{aligned}$$

D'où $\llbracket \lambda x.t \rrbracket \rightarrow \lambda x.t$. □

La fonction suivante extrait les λ -abstractions d'un λ -terme t et génère une substitution permettant de retrouver t .

Définition 10.14 (Abstraction des abstractions). *Pour toute position p , notons z_p une variable fraîche. Pour toute position p et tout λ -terme t , les équations inductives suivantes définissent une paire $\text{abs}_p(t) = (u, \sigma)$ formée d'un λ -terme u et d'une substitution σ .*

$$\begin{aligned} \text{abs}_p(x) &= (x, \emptyset) \\ \text{abs}_p(\lambda x.t) &= (z_p, \{z_p := \lambda x.t\}) \\ \text{abs}_p(t_1 t_2) &= (u_1 u_2, \sigma_1 \uplus \sigma_2) \quad \forall i \in \{1, 2\}, \text{abs}_{p.i}(t_i) = (u_i, \sigma_i) \end{aligned}$$

Lemme 10.15. *Soient une position p et un λ -terme t . Notons $\text{abs}_p(t) = (u, \sigma)$. Alors $t = u^\sigma$.*

Démonstration. Par induction sur t . □

Enfin, nous combinons la décomposition en squelettes avec l'extraction des λ -abstractions pour extraire les squelettes d'un λ -terme t . La substitution générée par l'abstraction est utilisée pour construire un unique terme cible.

Définition 10.16 (Abstraction des squelettes). *Considérons un λ -terme t , sa décomposition en squelettes $\llbracket t \rrbracket$, et l'abstraction des abstractions $\text{abs}_\epsilon(\llbracket t \rrbracket) = (u, \sigma)$. La **forme liftée** $\text{lft}(t)$ de t est le λ -terme défini ainsi :*

- Si $\sigma = \emptyset$, alors

$$\text{lft}(t) = (\lambda z_\epsilon. u)I$$
- Si $\sigma \neq \emptyset$, alors notons $\text{dom}(\sigma) = \{z_{p_1}, \dots, z_{p_n}\}$ et définissons

$$\text{lft}(t) = (\lambda z_{p_1} \dots z_{p_n}. u) \sigma(z_{p_1}) \dots \sigma(z_{p_n})$$

Considérer à part le cas d'une substitution vide permet d'assurer que quoi qu'il arrive $\text{lft}(t)$ a bien la forme d'une application, et peut donc être séparé en une fonction s_0 d'une part et une suite non-vide d'arguments $s_1 \dots s_n$ d'autre part.

Lemme 10.17 (Abstraction des squelettes). *Pour tout λ -terme t il existe une séquence de β -réduction faible*

$$\varrho : \text{lft}(t) \rightarrow t$$

Démonstration. Notons $\text{abs}_\epsilon(\llbracket t \rrbracket) = (u, \sigma)$ et raisonnons par cas sur σ .

- Si $\sigma = \emptyset$, alors $\text{lft}(t) = (\lambda z_e.u)I$ où la variable z_e n'apparaît pas dans t . De plus, la vacuité de σ implique que $\llbracket t \rrbracket$ ne contient pas de λ -abstraction, qui elle-même implique que t ne contient pas de λ -abstraction et que $u = t$. Donc $\text{lft}(t) \rightarrow t$.
- Si $\sigma \neq \emptyset$, alors $\text{lft}(t) = (\lambda z_{p_1} \dots z_{p_n}.u)\sigma(z_{p_1}) \dots \sigma(z_{p_n})$ pour $\text{dom}(\sigma) = \{z_{p_1}, \dots, z_{p_n}\}$.
Donc

$$\text{lft}(t) \rightarrow^n u^\sigma$$

Or $(u, \sigma) = \text{abs}_\epsilon(\llbracket t \rrbracket)$ donc par lemme 10.15 $u^\sigma = \llbracket t \rrbracket$. Enfin par lemme 10.13 $\llbracket t \rrbracket \rightarrow t$. Donc en composant ces séquences nous obtenons

$$\text{lft}(t) \rightarrow t$$

□

Remarquons de plus que pour tout t , le λ -terme $\text{lft}(t)$ est de la forme $s_0 s_1 \dots s_n$ où les s_i sont des formes normales. Nous pouvons enfin reprendre le cours de la preuve telle que décrite par Barendregt. Les formes normales t_a et t_b utilisées pour définir deux ensembles récursivement inséparables sont les λ -termes restreints $\lambda x_1 x_2 x_3. x_3$ et x_0 .

Lemme 10.18 (Adapté de Barendregt [Bar84]). *Soit F une stratégie optimale pour le λ -calcul faible. Soient s_0, s_1, \dots, s_n des λ -termes en forme normale. Soit $u = (\lambda x. x s_0 x)(\lambda y. y s_1 \dots s_n(II))$ où $I = \lambda z. z$ est le λ -terme identité. Alors*

- Si $s_0 s_1 \dots s_n \rightarrow \lambda x_1 x_2 x_3. x_3$
alors $F(u) = (\lambda y. y s_1 \dots s_n(II))s_0(\lambda y. y s_1 \dots s_n(II))$
- Si $s_0 s_1 \dots s_n \rightarrow x_0$
alors $F(u) = (\lambda x. x s_0 x)(\lambda y. y s_1 \dots s_n I)$

Démonstration. Similaire à [Bar84], lemme 13.5.5. □

Nous pouvons alors achever la preuve du théorème 10.6 dont l'énoncé est rappelé ici :

Théorème 10.6. *Il n'existe pas de stratégie optimale et calculable pour le λ -calcul faible.*

Démonstration. Par théorème 10.8 dérivé de Scott, les deux ensembles récursivement énumérables

$$\begin{aligned} \mathcal{A} &= \{ t \in \Lambda_r \mid t \rightarrow \lambda x_1 x_2 x_3. x_3 \} \\ \mathcal{B} &= \{ t \in \Lambda_r \mid t \rightarrow x_0 \} \end{aligned}$$

sont récursivement inséparables. Soit un entier $k \in \mathbb{N}$, notons $\text{lft}(\text{dec}(k)) = s_0 s_1 \dots s_n$ et définissons

$$u_k = (\lambda x. x s_0 x)(\lambda y. y s_1 \dots s_n(II))$$

Supposons qu'il existe une stratégie optimale F calculable. Alors, par lemme 10.18 l'ensemble

$$\mathcal{C}^* = \{ k \mid u_k \xrightarrow{\epsilon} F(u_k) \}$$

(où le ϵ dans $u_k \xrightarrow{\epsilon} F(u_k)$ spécifie que le radical réduit est à la racine) est calculable, et ainsi l'ensemble

$$\mathcal{C} = \{ \text{dec}(k) \mid k \in \mathcal{C}^* \}$$

sépare récursivement \mathcal{A} et \mathcal{B} , ce qui est impossible. \square

Ainsi, par le théorème 10.6 nous avons montré que les stratégies optimales sans partage ne pouvaient être calculables dans des CRS faibles orthogonaux contenant le λ -calcul faible. Nous pouvons par conséquent en déduire que la condition suffisante d'optimalité exprimée à la section 10.1 n'est pas décidable, point qui peut rappeler le fait que la nécessité justement ne soit pas décidable en λ -calcul non-faible [BKKS87]. À la section 10.3 suivante, nous allons voir que le partage pleinement paresseux en revanche permet d'obtenir plus facilement des séquences de réduction de même longueur que cet optimal inatteignable.

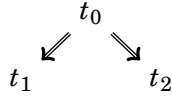
10.3 Partage et effectivité

Les sections 10.1 et 10.2 précédentes ont décrit pour les CRS orthogonaux faibles sans partage des stratégies de réduction qui sont toujours optimales, mais généralement non-calculables. La présente section s'applique comme la section 10.1 à tout CRS faible orthogonal, et démontre que les réductions nécessaires utilisant un partage pleinement paresseux tel que défini au chapitre 4 ont même longueur que les réductions nécessaires profondes sans partage évoquées à la section 10.1.

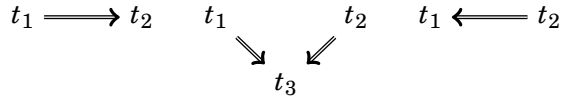
Définition 10.19 (Réduction nécessaire avec partage). *Une étape de **réduction nécessaire avec partage pleinement paresseux** est une étape de réduction parallèle étiquetée selon le chapitre 4 telle qu'au moins un des pas de réduction réduits en parallèle est nécessaire.*

Ainsi, il est démontré que les réductions nécessaires avec partage pleinement paresseux sont équivalentes aux réductions optimales sans partage. Cependant, alors que les réductions optimales sans partage sont généralement non-calculables, il existe des réductions nécessaires calculables dans de nombreux CRS faibles orthogonaux (auxquelles nous pouvons ajouter le partage pleinement paresseux). Nous obtenons donc l'effectivité des réductions faibles optimales, avec comme prix à payer le passage à la réduction partagée.

La première étape consiste à démontrer que, dans les CRS orthogonaux faibles, la réduction avec partage pleinement paresseux a la propriété du diamant si elle est restreinte aux pas de réduction nécessaires (lemme 10.22). Nous montrons pour cela que la réduction avec partage pleinement paresseux a la propriété de confluence forte (définition 0.13), qui peut être comprise comme une « propriété du diamant à effacement près » : tout diagramme



se ferme de l'une des trois manières suivantes :



Lemme 10.20. *La réduction partagée pleinement paresseuse dans un CRS orthogonal faible a la propriété de confluence forte.*

Démonstration. Similaire au lemme 7.51 de permutation. \square

La propriété de confluence forte se transforme en une vraie propriété du diamant quand les pas de réduction observés ne sont pas effaçables. Ceci s'applique en particulier aux pas de réduction nécessaires, qui sont connus pour leur caractère non-effaçable rappelé par le lemme suivant [GK96].

Lemme 10.21 (Ineffaçabilité des pas de réduction nécessaires). *Considérons un ARS muni d'une opération de résiduation. Soit ρ_n un pas de réduction nécessaire de source o normalisable. Soit ρ un pas de réduction de source o différent de ρ_n . Alors ρ_n admet au moins un résidu après ρ (et au moins un de ces résidus est nécessaire).*

La combinaison des lemmes 10.20 de confluence forte de la réduction partagée pleinement paresseuse des CRS orthogonaux faibles et 10.21 d'ineffaçabilité des pas de réduction nécessaires nous assure que dans tout CRS orthogonal faible la réduction nécessaire avec partage pleinement paresseux a la propriété du diamant. En particulier, dans un tel système et pour un terme de départ donné t , toutes les séquences de réduction normalisantes nécessaires avec partage pleinement paresseux ont même longueur.

Lemme 10.22. *Dans tout CRS faible orthogonal les séquences de réduction normalisantes nécessaires avec partage pleinement paresseux ont toutes même longueur.*

Nous obtenons le résultat final de cette section, qui est l'équivalence de la réduction sans partage optimale avec la réduction nécessaire avec partage pleinement paresseux, en vérifiant que la réduction nécessaire profonde sans partage est une instance de la réduction nécessaire avec partage pleinement paresseux. En effet, la réduction nécessaire profonde ne duplique aucun pas de réduction nécessaire et la réduction nécessaire profonde sans partage est donc confondue avec la réduction nécessaire profonde avec partage pleinement paresseux.

Lemme 10.23. *Considérons un terme t dans un CRS faible, et deux pas de réduction ρ_o et ρ_i de source t tels que la position de ρ_o est un préfixe strict de la position de ρ_i (ainsi ρ_i est plus profond que ρ_o). Alors ρ_o a au plus un résidu après ρ_i .*

Démonstration. La position p_o de ρ_o est un préfixe strict de la position p_i de ρ_i . Donc l'ensemble des descendants de la position p_o après ρ_i est le singleton $\{p_o\}$. Comme tout résidu ρ'_o de ρ_o a une position p'_o qui est un descendant de p_o , le pas de réduction ρ_o ne peut pas avoir plus d'un résidu. \square

Nous obtenons finalement le théorème suivant d'équivalence de longueur entre les réductions nécessaires profondes sans partage et les réductions nécessaires avec partages pleinement paresseux.

Théorème 10.24. *Considérons un terme normalisable t dans un CRS orthogonal faible. Soient ρ_1 une séquence de réduction nécessaire profonde sans partage normalisante à partir de t et $\bar{\rho}_2$ une séquence de réduction nécessaire avec partage pleinement paresseux normalisante à partir de t . Alors ρ_1 et $\bar{\rho}_2$ ont même longueur.*

Démonstration. Considérons que la séquence ρ_1 s'applique dans le CRS étiqueté $R(\mathcal{R})^l$ définissant le partage pleinement paresseux selon le chapitre 4 du CRS Σ considéré. Par lemme 10.23 la séquence ρ_1 est une séquence de réduction parallèle étiquetée $\bar{\rho}_1$. Cette séquence étant de plus nécessaire, par lemme 10.22 elle a même longueur que $\bar{\rho}_2$. \square

Ainsi, nous apprenons deux choses sur la comparaison des réductions sans partage et des réductions avec partage pleinement paresseux dans les CRS orthogonaux faibles. D'abord, les deux modes de réduction définissent des espaces de réduction dont les séquences normalisantes les plus courtes ont même longueur, ce qui signifie en particulier que le partage pleinement paresseux ne permet jamais d'obtenir une séquence de réduction strictement plus courte que la séquence optimale sans partage. En revanche, les séquences de réduction atteignant cette longueur optimale sont beaucoup plus nombreuses dans le mode partagé. Avec partage les duplications sont neutralisées, et les pas de réduction nécessaires peuvent être réduits dans un ordre arbitraire. Il suffit ainsi à chaque étape de réduction d'être capable de détecter *un* pas de réduction nécessaire pour obtenir une séquence optimale. Or, dans de nombreux systèmes séquentiels tels que le λ -calcul, le radical « le moins profond » ou « le plus à gauche » est systématiquement nécessaire.

C'est ainsi en particulier que la technique de réduction de Wadsworth [Wad71], qui sélectionne toujours un β -radical le plus à gauche, fournit des séquences de réduction de graphes dont la longueur est exactement la longueur des plus courtes séquences de β -réduction faible sans partage. Ceci formalise donc un sens dans lequel la technique de Wadsworth est optimale.

10.4 Un cadre axiomatique pour les réductions optimales de familles

Avec la section 10.3 précédente nous avons obtenu un premier résultat d'optimalité pour la réduction pleinement paresseuse dans les CRS faibles

orthogonaux : la longueur de toute séquence de réduction nécessaire avec partage pleinement paresseux normalisante est égale à la longueur de la plus courte séquence de réduction sans partage normalisante. L'étape suivante consiste à démontrer que la réduction nécessaire avec partage pleinement paresseux n'est pas optimale seulement dans l'espace des réductions faibles sans partage, mais également dans l'espace des réductions de *familles faibles*.

Pour prouver ce résultat d'optimalité parallèle (théorème 10.41), nous allons utiliser un cadre axiomatique introduit par Glauert et Khasidashvili généralisant les concepts de l'évaluation optimale à la Lévy aux systèmes abstraits de réécriture [GK96].

Nous pouvons résumer la partie de leur travail qui nous est utile ici aux points suivants :

1. Prenons un système de résiduation déterministe (*DRS*), c'est-à-dire essentiellement un *ARS* muni d'une opération de résiduation vérifiant la propriété des développements finis (définition 0.18),
2. nous en déduisons d'une part une notion de réduction nécessaire, et d'autre part une notion de zig-zag généralisant celle de Lévy.
3. Ajoutons une notion de famille compatible avec le zig-zag susnommé et vérifiant la propriété des développements de familles finis,
4. nous en déduisons l'optimalité de la *réduction nécessaire complète* de familles parmi l'ensemble des réductions de familles.

Cette construction de Glauert et Khasidashvili est encore paramétrée par un ensemble cible vérifiant lui aussi certains axiomes. Pour simplifier ici nous prenons comme ensemble cible l'ensemble des formes normales, pour lequel ces axiomes supplémentaires sont triviaux.

La suite de cette section rappelle les définitions et résultats de Glauert et Khasidashvili [GK96] et précise leur portée. Le point principal est la caractérisation des familles de pas de réduction d'un *DRS*, qui fixent les pas de réduction que nous sommes autorisés à effectuer en parallèle. Ceci définit en particulier l'espace de réductions dans lequel le résultat d'optimalité sera valide.

Ces *familles* de pas de réduction légitimement parallélisables sont déduits d'une version abstraite de la condition dite de *zig-zag*, qui est un des outils introduits par Lévy pour décrire les familles de radicaux dans le λ -calcul [Lé78].

La relation de zig-zag (définition 10.29) est basée sur un critère de descendance. Ainsi, comme discuté en introduction, l'appartenance de deux radicaux à une même famille dépend des séquences de réduction passées. Les familles (et le zig-zag) sont donc formellement des relations d'équivalence sur les pas de réduction *avec histoire* (définition 10.25).

Définition 10.25 (Pas de réduction avec histoire). *Un pas de réduction avec histoire est une paire (ρ, ρ) où ρ est une séquence de réduction, ρ est un pas de réduction, et $\text{tgt}(\rho) = \text{src}(\rho)$.*

Convention 10.26. *Rappelons que dans un système de résiduation déterministe DRS (définition 0.18), tout les développements complets d'un ensemble R de*

pas de réduction co-initiaux sont équivalents, et nous désignons par $\text{dev}(R)$ l'un quelconque de ces développements complets.

Dans ce cadre, nous pouvons encore définir une opération de résiduation sur les séquences de réduction, dans laquelle le résidu d'une séquence de réduction n'est plus un ensemble de pas de réduction co-initiaux mais une autre séquence définie par l'équation

$$(\rho_1 \cdot \rho) / \rho_2 = \text{dev}(\rho_1 / \rho_2) \cdot (\rho / \text{dev}(\rho_2 / \rho_1))$$

où ρ_1 et ρ_2 sont des pas de réduction et ρ une séquence de réduction.

Définition 10.27 (Équivalence par permutation). *La relation d'équivalence par permutation \equiv_P est la plus petite relation d'équivalence sur les séquences de réduction telle que les deux conditions suivantes sont vérifiées :*

- Pour tous ensembles R_1 et R_2 de pas de réductions co-initiaux

$$\text{dev}(R_1) \cdot \text{dev}(R_2 / \text{dev}(R_1)) \equiv_P \text{dev}(R_2) \cdot \text{dev}(R_1 / \text{dev}(R_2))$$

- Pour toutes séquences de réduction ρ_1, ρ_2, ρ'_2 , et ρ_3 telles que $\text{tgt}(\rho_1) = \text{src}(\rho_2) = \text{src}(\rho'_2)$ et $\text{src}(\rho_3) = \text{tgt}(\rho_2) = \text{tgt}(\rho'_2)$, si $\rho_2 \equiv_P \rho'_2$ alors

$$\rho_1 \cdot \rho_2 \cdot \rho_3 \equiv_P \rho_1 \cdot \rho'_2 \cdot \rho_3$$

Remarque 10.28. *La relation d'équivalence par permutation est réputée équivalente à la condition suivante de résiduation [GK96] :*

$$\rho_1 \equiv_P \rho_2 \quad \text{si et seulement si} \quad \rho_1 / \rho_2 = \rho_2 / \rho_1 = \emptyset$$

Définition 10.29 (Zig-zag). *Soient deux pas de réduction avec histoire (ρ_1, ρ_1) et (ρ_2, ρ_2) avec ρ_1 et ρ_2 des séquences de réduction co-initiales. Nous disons que (ρ_2, ρ_2) est une copie de (ρ_1, ρ_1) si les deux conditions suivantes sont vérifiées :*

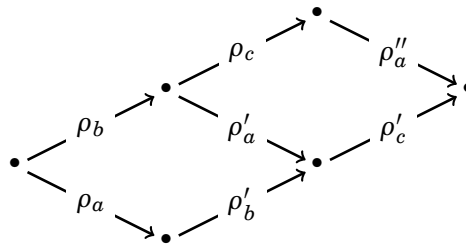
- $\rho_1 \cdot (\rho_2 / \rho_1) \equiv_P \rho_2$
- $\rho_2 \in \rho_1 / (\rho_2 / \rho_1)$

La relation de **zig-zag** est la clôture symétrique et transitive de la relation de copie (c'est-à-dire la plus petite relation d'équivalence contenant le relation de copie).

Exemple 10.30.

Considérons dans l'espace de réduction figuré ci-dessous que :

- ρ'_a est le résidu de ρ_a après ρ_b ,
- ρ''_a est le résidu de ρ'_a après ρ_c ,
- ρ'_b est le résidu de ρ_b après ρ_a , et
- ρ'_c est le résidu de ρ_c après ρ'_a .



Alors, par exemple :

- $(\rho_b \rho_c, \rho''_a)$ est une copie de (ϵ, ρ_a) et de (ρ_b, ρ'_a) .
- $(\rho_b \rho'_a, \rho'_c)$ est une copie de (ρ_b, ρ_c) .

Une *structure de familles déterministe* (*Deterministic Family Structures*, *DFS*) est un *DRS* muni d'une notion de familles définissant les réductions parallèles possibles. Cette structure est basée sur la relation de zig-zag et fait référence aux notions de causalité discutées dans la partie III.

Définition 10.31 (*Deterministic Family Structures (DFS)*). Soit un quadruplet $(\Sigma, /, \simeq, \hookrightarrow)$ où $(\Sigma, /)$ est un *DRS*, \simeq est une relation d'équivalence sur les pas de réduction avec histoire, et \hookrightarrow est une relation sur les classes d'équivalence de \simeq . Pour tout pas de réduction avec histoire (ρ, ρ) , notons $\phi = \text{Fam}(\rho, \rho)$ la classe d'équivalence de (ρ, ρ) pour \simeq , appelée **famille** de (ρ, ρ) .

Le quadruplet $(\Sigma, /, \simeq, \hookrightarrow)$ est un *DFS* si les axiomes suivants sont vérifiés.

Zig-zag La relation \simeq contient le zig-zag.

Familles initiales Pour tous deux pas de réduction co-initiaux distincts ρ_1 et ρ_2 nous avons $\text{Fam}(\epsilon, \rho_1) \neq \text{Fam}(\epsilon, \rho_2)$.

Contribution Pour toutes deux familles ϕ_1 et ϕ_2 , nous avons $\phi_1 \hookrightarrow \phi_2$ si et seulement si pour tout pas de réduction avec histoire $(\rho_2, \rho_2) \in \phi_2$ il existe un pas de réduction avec histoire $(\rho_1, \rho_1) \in \phi_1$ tel que $\rho_1 \cdot \rho_1 < \rho_2$ (où $<$ est la relation préfixe sur les séquences).

Création Pour tout pas de réduction avec histoire (ρ, ρ) , si ρ crée un pas de réduction ρ' alors $\text{Fam}(\rho, \rho) \hookrightarrow \text{Fam}(\rho \cdot \rho, \rho')$.

Terminaison Toute séquence de réduction contenant des pas de réduction d'un nombre fini de familles termine.

Définition 10.32 (Étape de réduction de famille). Une **étape de réduction de famille** est une séquence de réduction $\rho : t \rightarrow t'$ dont tous les éléments appartiennent à une même famille ϕ . Cette étape ρ est **complète** si de plus la cible t' n'est la source d'aucun pas de réduction de la famille ϕ .

Définition 10.33 (Famille nécessaire). Dans un *DFS*, une **famille nécessaire** ϕ est une famille telle qu'il existe un pas de réduction avec histoire $(\rho, \rho) \in \phi$ tel que le pas de réduction ρ est nécessaire.

Théorème 10.34 (Optimalité [GK96]). Soit un terme normalisable t dans un *DFS*, et $\bar{\rho}$ une séquence de réduction de familles normalisante de source t dont chaque étape réduit complètement une famille nécessaire. Alors aucune séquence de réduction de familles normalisante de source t n'est strictement plus courte que $\bar{\rho}$.

Ainsi dans tout *DFS* une séquence de réduction de familles $\bar{\rho}$ est optimale si toute étape $\bar{\rho}$ vérifie deux conditions :

- L'étape parallèle $\bar{\rho}$ réduit tous les membres d'une famille présents dans sa source.

- La famille concernée par l'étape parallèle $\bar{\rho}$ contient au moins un pas de réduction nécessaire.

Notons également que cette optimalité est exprimée dans l'ensemble des séquences de réduction de familles, c'est-à-dire qu'elle concerne des séquences de réduction parallèle qui ne peuvent jamais réduire en une étape plus de choses qu'une famille complète.

Nous pourrions donc résumer la situation par l'énoncé « les réductions de familles complètes nécessaires sont optimales parmi les réductions qui ne partagent pas plus qu'elles ». Ainsi il est évident que les réductions de familles complètes nécessaires font toujours des choix localement optimaux, d'abord en sélectionnant des pas de réduction nécessaires et ensuite en réduisant entièrement les familles associées.

Dans ces conditions, et bien que les analogies avec des situations dans lesquelles des enchaînements de choix localement optimaux sont loin de fournir des solutions globales optimales ne manquent pas, il est légitime de se demander si le théorème 10.34 apporte réellement quelque chose.

Ce théorème a en fait un enjeu principal, résumé par la question « est-il possible que des membres d'une même famille ϕ apparaissent à des instants différents dans une séquence de réduction $\bar{\rho}$? » Dans une telle situation, commencer à réduire cette famille ϕ avant que tous ses membres ne soient créés pourrait demander de réduire plusieurs fois la famille ϕ , ce qui est une source potentielle de non-optimalité. Il vaudrait donc mieux attendre que tous les membres de ϕ soient créés avant d'agir, pour réduire ensuite la famille en une seule fois. Seulement il n'est pas évident que cela soit possible, et nous pouvons en particulier imaginer une famille ϕ dans laquelle un pas de réduction $\rho_1 \in \phi$ est nécessaire à la création d'un autre pas de réduction $\rho_2 \in \phi$.

Les théorèmes d'optimalité de Lévy [Lé78, Lé80] et de Glauert et Khasidashvili [GK96] assurent donc que dans une séquence de réduction nécessaire complète chaque famille est réduite une fois pour toute, et qu'ainsi le choix localement optimal de « réduire tout de suite le plus possible parmi ce qui est nécessaire » est aussi globalement optimal. Nous allons maintenant montrer que ceci s'applique à notre réduction avec partage pleinement paresseux.

10.5 Optimalité faible

Comme dans tout cadre axiomatique, l'applicabilité de ce résultat est limitée par deux difficultés principales : donner une contrepartie concrète aux notions abstraites, et vérifier que les notions concrètes vérifient les axiomes.

En considérant les systèmes du premier ordre orthogonaux étiquetés qui sont les cibles des opérations de lifting présentées aux chapitres 8 et 9 toutes ces difficultés disparaissent face aux remarques suivantes :

- La notion habituelle de résidus dans les systèmes du premier ordre orthogonaux vérifie la propriété des développements finis et définit un *DRS* [GK96].

- Les étiquettes contiennent tout ce qu'il faut savoir sur l'histoire d'un pas de réduction avec histoire, et fournissent une définition simple des familles et de la relation de contribution. De plus les propriétés causales déjà démontrées à la section 7.5 suffisent à assurer les axiomes de zig-zag et de contribution.
- Les systèmes du premier ordre orthogonaux vérifient la propriété de développement de familles finis [vO97].

Formalisons maintenant ces remarques pour obtenir le résultat d'optimalité de la réduction pleinement paresseuse. Nous commençons par exhiber le système étiqueté utilisé pour définir la réduction partagée.

Définition 10.35. *Considérons Σ un CRS orthogonal faible et (\mathcal{F}, ψ) un système de lifting stable pour Σ . Notons $\Sigma^{\mathcal{L}}$ l'étiquetage causal selon le chapitre 7 (avec raffinement canonique et expansion), et $(\mathcal{F}^{\mathcal{L}}, \psi^{\mathcal{L}})$ le système de lifting stable pour $\Sigma^{\mathcal{L}}$ préservant le partage donné par la section 9.4. Enfin, notons $\Sigma_f^{\mathcal{L}}$ le système du premier ordre orthogonal étiqueté défini par la réduction cible.*

Lemme 10.36. *Le système $\Sigma_f^{\mathcal{L}}$ muni de la notion de résiduation usuelle / (définition 0.78) est un DRS.*

Démonstration. Par orthogonalité du CRS d'origine Σ et théorème 9.37, le système $\Sigma_f^{\mathcal{L}}$ est un système de réécriture du premier ordre orthogonal. Il est donc un DRS [GK96]. \square

Dans cette section nous considérons les pas de réduction avec histoire (ρ, ρ) dans $\Sigma_f^{\mathcal{L}}$ tels que la source $\text{src}(\rho)$ un terme initial. Nous pouvons définir sur ces pas de réduction avec histoire une relation d'équivalence basée sur les étiquettes.

Définition 10.37. *La relation d'équivalence \simeq sur les pas de réduction avec histoire est définie par la condition $(\rho_1, \rho_1) \simeq (\rho_2, \rho_2)$ si et seulement si les pas de réduction ρ_1 et ρ_2 ont même nom.*

Remarque 10.38. *Nous ne nous servons pas des histoires ρ_1 et ρ_2 dans cette caractérisation, car le principe de l'étiquetage causal est que les étiquettes de la cible de la séquence de réduction ρ_i (c'est-à-dire de la source du pas de réduction ρ_i) contiennent déjà toutes les informations pertinentes à propos de cette séquence.*

Ainsi chaque famille de pas de réduction avec histoire est caractérisée par un nom de radical. Nous avons déjà défini aux chapitres 5, 6, et 7 une relation de contribution \hookrightarrow sur les noms de radicaux, que nous réutilisons ici et qui induit une relation de contribution encore notée \hookrightarrow sur les familles de pas de réduction avec histoire. Rappelons la définition de cette relation ainsi que des étiquettes (définitions 7.14 et 7.24).

Définition 10.39. *Les étiquettes α et les noms de pas de réduction Ω sont définis par la grammaire suivante :*

$$\begin{aligned} \alpha & ::= p \mid [\Omega, \alpha] && p \text{ position} \\ \Omega & ::= \alpha_1 \dots \alpha_n && n \geq 1 \end{aligned}$$

La relation de contribution directe \hookrightarrow^d est la plus petite relation sur les noms de pas de réduction telle que pour tout nom de pas de réduction Ω , pour toute étiquette α et pour toutes séquences d'étiquettes éventuellement vides Γ_1 et Γ_2 nous avons :

$$\Omega \hookrightarrow^d \Gamma_1[\Omega, \alpha]\Gamma_2$$

La relation de contribution \hookrightarrow est la clôture transitive de \hookrightarrow^d .

Rappelons également que la relation de contribution \hookrightarrow vérifie des propriétés de correction et de complétude causales (théorèmes 7.50 et 7.52) vis-à-vis de la réduction source étiquetée. Ces propriétés sont encore valables avec la réduction cible étiquetée, car source et cible définissent la même relation \hookrightarrow^n d'antécédent nécessaire (définition 7.45), par stabilité de la nécessité par *lifting* (similaire au lemme 10.3).

Nous pouvons maintenant conclure ce chapitre en vérifiant que les structures ainsi définies vérifient les axiomes des *DFS*.

Théorème 10.40. *Le système $(\Sigma_f^{\mathcal{L}}, /, \simeq, \hookrightarrow)$ est un DFS.*

Démonstration.

Zig-zag Si un pas de réduction avec histoire (ρ_2, ρ_2) est une copie d'un pas de réduction avec histoire (ρ_1, ρ_1) , alors en particulier $\rho_2 \in \rho_1 / (\rho_2 / \rho_1)$. Donc par stabilité des radicaux (lemme 7.46) les pas de réduction ρ_2 et ρ_1 ont même nom, et sont dans la même famille. Donc \simeq est une relation d'équivalence contenant la relation de copie, et ainsi elle contient le zig-zag.

Familles initiales Soient deux pas de réduction co-initiaux ρ_1 et ρ_2 dont la source commune est un terme initial. Alors si ρ_1 et ρ_2 sont différents en particulier ils s'appliquent à des positions différentes et n'ont pas la même étiquette racine. Donc leurs noms sont différents et $\text{Fam}(\epsilon, \rho_1) \neq \text{Fam}(\epsilon, \rho_2)$.

Contribution Avec le vocabulaire du chapitre 7 cet axiome peut être reformulé pour se ramener à l'égalité entre la relation de contribution \hookrightarrow et la relation de nécessité \hookrightarrow^n (définition 7.45). Cette égalité est prouvée par les théorèmes 7.50 et 7.52 de correction et de complétude causale.

Création Application directe du lemme 7.40 de contribution directe.

Terminaison Les systèmes du premier ordre orthogonaux vérifient la propriété de développement de familles finis [vO97]. \square

Ainsi, le théorème 10.34 s'applique aux réductions nécessaires partagées pleinement paresseuses dans les *CRS* faibles orthogonaux, et montre l'optimalité de ces réductions par rapport à toute séquence de réduction partagée dans laquelle le partage est justifié causalement.

Théorème 10.41 (Optimalité faible). *Soit t un terme normalisable dans un CRS faible orthogonal, et $\bar{\rho}$ une séquence de réduction de familles faibles complètes nécessaires normalisante de source t . Alors aucune séquence de réduction de familles faibles normalisante de source t n'est strictement plus courte que $\bar{\rho}$.*

Bilan du chapitre

Dans ce chapitre nous avons récolté les fruits de l'accumulation de constructions proposées tout au long de cette thèse pour les *CRS* faibles, et en particulier pour les *CRS* faibles orthogonaux.

Nous avons en particulier démontré que la réduction nécessaire avec partage pleinement paresseux définie au chapitre 4 était de longueur égale à la réduction faible sans partage optimale, et que cette réduction avec partage pleinement paresseux était en outre optimale parmi les réductions de familles faibles où les familles sont déduites des notions de causalité analysées au chapitre 7.

En retournant ces résultats nous apprenons également que dans les *CRS* faibles orthogonaux, il est impossible de justifier causalement un partage qui permette des séquences de réduction plus courtes que les réductions optimales sans partage. Cependant, l'utilisation du partage permet dans de nombreux cas d'obtenir des stratégies de réduction calculables produisant des séquences de réduction de longueur minimale, alors que les stratégies optimales sans partage ne sont pas calculables.

En retournant ces résultats sous un autre angle encore, nous en déduisons que pour faire mieux que la pleine paresse il faut s'affranchir de la réduction faible ou relâcher les justifications causales du partage. Nous allons voir dans le prochain chapitre que les outils présentés dans cette thèse, et en particulier les systèmes de partage étiquetés *SLS* permettent tout ceci.

CHAPITRE 11

Au-delà de l’optimalité faible

Sommaire

11.1 Sous-optimalité des familles faibles	330
11.2 Partage optimisé	333
11.3 Partage en réduction non-faible	337
11.4 Réduction (non-)faible et (non-)orthogonalité	340
11.5 Zones d’effet optimisées	343
11.6 Un dernier exemple	344

Du chapitre 10 précédent émerge une notion unifiée d’optimalité faible pour les *CRS* orthogonaux, correspondant à trois critères :

- réduction faible sans partage de longueur minimale,
- réduction faible nécessaire avec partage pleinement paresseux, et
- réduction de familles de Lévy faibles nécessaires.

Cette optimalité faible a pour caractéristique de pouvoir être mise en œuvre grâce à un partage de sous-termes qui peut être spécifié de manière simple par les systèmes de partage étiquetés *SLS* du chapitre 1. Nous allons maintenant voir comment, sans quitter le monde des *SLS* et du partage de sous-termes, il est possible d’obtenir des séquences de réduction de longueur inférieure à cette borne d’optimalité faible. Ces constructions reposeront sur trois pistes : introduction de partage dans la zone d’effet d’un pas de réduction (section 11.2), diminution de l’étendue des zones d’effet (section 11.5), et support des réductions non-faibles (section 11.3).

Dans un premier temps nous resterons en réduction faible, et verrons que dans les *CRS* faibles le critère de *zig-zag* ignore certains partages de pas de réduction qui « à l’œil nu » pourraient sembler légitimes (section 11.1). Pour répondre à cette remarque, un nouvel étiquetage des *CRS* faibles (section 11.2) donnera un moyen de partager plus de sous-termes que le partage pleinement paresseux du chapitre 4.

Nous verrons ensuite que la réduction non-faible, qui avait été essentiellement bannie dans les chapitres précédents¹, s’intègre très bien à notre forma-

1. Les opérations de *méta-lifting* des chapitres 8 et 9 sont la seule exception à la réduction

lisme (section 11.3), et nous discuterons à la section 11.4 deux choses qu'apporte concurremment la réduction non-faible : non-orthogonalité et possibilité d'optimisation des stratégies de réduction avec partage de sous-termes.

Enfin la section 11.5 rappellera comment l'équivalence des différents modèles du partage pleinement paresseux établie au chapitre 3 dépendait de la réduction faible, et donc comment la possibilité de réduction non-faible permet de tirer parti d'approches de la pleine paresse autres que celle de Wadsworth, puis la section 11.6 proposera un système étiqueté concret qui rassemble les idées de ce chapitre afin de proposer un mécanisme de réduction optimisé pour le λ -calcul.

11.1 Sous-optimalité des familles faibles

Le partage pleinement paresseux défini au chapitre 4 et le critère de *zig-zag* en réduction faible du chapitre 10 ignorent certains partages de sous-termes que nous pourrions vouloir réaliser.

Exemple 11.1.

Considérons le CRS donné par les deux règles suivantes :

$$\begin{aligned}\kappa_d &: @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0)) \\ \kappa_f &: f(Z, Z) \rightarrow h(Z, Z)\end{aligned}$$

et observons le terme

$$t = @(\langle x \rangle f(x, x), a)$$

Le terme t contient un pré-radical $f(x, x)$ pour la règle κ_f , qui est gelé par la présence de la variable x qui est liée par le contexte $@(\langle x \rangle, \square, a)$. Nous avons en revanche un pas de réduction ρ_d avec la règle κ_d , qui a pour cible

$$t' = f(f(a, a), f(a, a))$$

Dans le terme t' apparaissent simultanément trois radicaux pour la règle κ_f , un pour chaque occurrence de f , définissant trois pas de réduction ρ_f^ϵ , ρ_f^1 , et ρ_f^2 .

Aucun des trois radicaux créés à l'exemple 11.1 n'est dans la même famille faible qu'un autre. Nous pouvons en effet voir que la condition de *zig-zag* ne peut pas s'appliquer, puisque le pas de réduction ρ_d est le seul moyen de créer l'un quelconque de ρ_f^ϵ , ρ_f^1 , ou ρ_f^2 . Ceci est d'ailleurs confirmé par l'exemple 11.2 suivant qui montre l'étiquetage pleinement paresseux du pas de réduction ρ_d , c'est-à-dire un étiquetage spécifiant une notion de réduction parallèle équivalente à la réduction de familles complètes.

Exemple 11.2.

Considérons l'étiquetage suivant du terme t de l'exemple 11.1

$$@^{\omega}(\langle x \rangle^{\alpha} f^{\beta}(x, x), a^{\gamma})$$

faible rencontrée jusqu'ici.

La réduction étiquetée selon le chapitre 4 utilise la règle étiquetée

$$\kappa_d^l : @^\omega(\langle x \rangle^\alpha f^\beta(x, x), Z_0) \rightarrow f^\epsilon(f^1(Z_0, Z_0), f^2(Z_0, Z_0))$$

obtenue à partir de la règle raffinée venant de la valuation partielle

$$\{ Z \mapsto \underline{\lambda}x_1.f(x_1, x_1) \}$$

La règle étiquetée κ_d^l est utilisée dans le contexte vide et avec la valuation

$$\{ Z_0 \mapsto \underline{\lambda}\phi.a^\gamma \}$$

Cela forme un pas de réduction ρ_d^l dont la cible est

$$f^{[\omega, \epsilon]}(f^{[\omega, 1]}(a^\gamma, a^\gamma), f^{[\omega, 2]}(a^\gamma, a^\gamma))$$

dans lequel les quatre occurrences du symbole a sont partagées sous l'effet de l'étiquette γ et les trois occurrences du symboles f sont dissociées par les étiquettes $[\omega, \epsilon]$, $[\omega, 1]$, et $[\omega, 2]$. En particulier, il n'y a aucun partage des trois pas de réduction créés pour la règle κ_f .

Cependant, les trois pas de réduction ρ_f^ϵ , ρ_f^1 , et ρ_f^2 de l'exemple 11.1 ont visiblement quelque chose en commun : les trois radicaux qui les définissent descendent tous de la même unique occurrence du pré-radical gelé $f(x, x)$ et sont simultanément dégelés (et donc créés en tant que radicaux) par le seul pas de réduction ρ_d . Ce fait n'est pas visible dans un cadre abstrait comme celui du zig-zag, qui ne tient compte que de la résiduation et donc que des radicaux (qui ne sont pas gelés). En revanche, dans notre cadre de termes nous possédons également des informations de descendance, qui permettent de suivre également les pré-radicaux gelés. Et de la même manière que le zig-zag légitime le partage de deux radicaux ayant un même radical ancêtre (par résiduation), nous pouvons considérer comme légitime le partage de deux radicaux ayant un même pré-radical ancêtre (par descendance).

De manière plus générale, si deux sous-termes t_1 et t_2 ont un ancêtre commun t_0 (par descendance) alors nous avons affaire à une duplication, et le partage de t_1 et t_2 est justifié. Ne reste qu'à voir si nous sommes capables de le généraliser. C'est d'ailleurs justement ce qu'a l'air de faire le partage pleinement paresseux : le lemme 9.45 liant descendance et étiquettes a montré sur un étiquetage équivalent à l'étiquetage pleinement paresseux que les étiquettes reflétaient toutes les relations de descendance. Interprété en termes de partage, ce résultat signifie que deux descendants d'un même sous-terme sont systématiquement partagés. Comment se fait-il alors que l'étiquetage pleinement paresseux ne prescrive pas plus de partage à l'exemple 11.2?

L'explication vient du raffinement canonique utilisé par l'étiquetage, qui, comme le montre l'exemple 11.3 suivant, transforme les règles d'une manière qui efface certaines informations de descendance, et en particulier les informations de descendance qui concernent les nouvelles copies de fragments du terme source.

Exemple 11.3.

Rappelons la règle κ_d de l'exemple 11.1 et son raffinement κ_d^r utilisé à l'exemple 11.2.

$$\begin{aligned}\kappa_d & : @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0)) \\ \kappa_d^r & : @(\langle x \rangle f(x, x), Z_0) \rightarrow f(f(Z_0, Z_0), f(Z_0, Z_0))\end{aligned}$$

Ces deux règles permettent de définir deux pas de réduction ρ_d et ρ_d^r de même source et de même cible.

$$\begin{aligned}\rho_d & : @(\langle x \rangle f(x, x), a) \rightarrow f(f(a, a), f(a, a)) \\ \rho_d^r & : @(\langle x \rangle f(x, x), a) \rightarrow f(f(a, a), f(a, a))\end{aligned}$$

Les pas ρ_d et ρ_d^r diffèrent en revanche par leur relation de résiduation :

$$\begin{array}{l|l} 2/\rho_d = \{11, 12, 21, 22\} & 2/\rho_d^r = \{11, 12, 21, 22\} \\ 11/\rho_d = \{\epsilon, 1, 2\} & 11/\rho_d^r = \emptyset \end{array}$$

Pour résumer l'exemple 11.3, notons deux faits sur la règle raffinée. D'abord, la règle raffinée préserve la descendance sur les occurrences du symbole a , qui n'est pas concerné par le raffinement. Ceci est également visible à l'exemple 11.2 avec la préservation de l'étiquette γ . Ensuite, la règle raffinée supprime la descendance sur les occurrences du symbole f . En effet, une fois le raffinement appliqué, les occurrences de f dans le membre droit apparaissent comme nouvellement créées. Ceci est de même visible à l'exemple 11.2 avec la création de trois nouvelles étiquettes distinctes. Et finalement, en faisant perdre des informations sur la descendance, le raffinement fait également perdre des opportunités de partage.

Vient alors la question de la modification du système. Pouvons-nous avoir un nouveau système d'étiquettes pour les CRS faibles qui permette de partager des pas de réductions tels que les ρ_f^ϵ , ρ_f^1 , et ρ_f^2 de l'exemple 11.1 ?

Permettons-nous ici une réponse prudente et nuancée : « oui et non ».

En effet, remarquons premièrement que partager par exemple les pas de réduction ρ_f^ϵ et ρ_f^1 ne serait pas possible dans le cadre des SLS. Ceci demanderait en effet dans le terme $f(f(a, a), f(a, a))$ de donner la même étiquette aux occurrences de f des deux positions ϵ et 1. Ceci casserait donc inévitablement la propriété de partage, puisque les deux radicaux $f(f(a, a), f(a, a))$ et $f(a, a)$ ne sont pas égaux en tant que sous-termes. Cette impossibilité peut aussi être vue dans le fait qu'un tel partage introduirait un cycle dans le graphe, ce qui de manière générale n'est pas expressible dans notre formalisme des SLS.

En revanche, cette remarque ne s'applique pas au partage des pas de réduction ρ_f^1 et ρ_f^2 , dont les deux radicaux sont égaux à $f(a, a)$. Ainsi, il n'est pas interdit d'imaginer une règle de réduction étiquetée de la forme suivante, dans laquelle l'étiquetage du membre droit n'est pas l'étiquetage initial positionnel que nous avons toujours utilisé jusqu'ici, mais fait apparaître à deux endroits la même étiquette 1.

$$@^\omega(\langle x \rangle^\alpha f^\beta(x, x), Z_0) \rightarrow f^\epsilon(f^1(Z_0, Z_0), f^1(Z_0, Z_0))$$

Par cette règle nous pourrions avoir un pas de réduction étiqueté

$$@^\omega(\langle x \rangle^\alpha f^\beta(x, x), a^\gamma) \rightarrow f^{[\omega, \epsilon]}(f^{[\omega, 1]}(a^\gamma), f^{[\omega, 1]}(a^\gamma))$$

dans la cible duquel les deux occurrences du radical $f^{[\omega, 1]}(a^\gamma)$ sont partagées.

À la section 11.2 suivante nous formalisons une nouvelle technique d'étiquetage pour les *CRS* faibles qui introduit un tel partage.

11.2 Partage optimisé

Dans les chapitres 4, 7 et 9 nous avons manipulé des étiquetages des *CRS* qui étaient « internalisés » dans le formalisme des *CRS*. Ceci donnait une certaine concision au développement formel et permettait d'utiliser sur les systèmes étiquetés les constructions et les résultats développés pour les *CRS* généraux, comme le *méta-lifting* au chapitre 9.

Cependant cette internalisation des étiquettes requiert un raffinement préalable des systèmes, transformation qui n'a pas d'impact sur les réductions mais qui allonge le chemin à parcourir entre un *CRS* donné et sa version avec partage. Notamment, l'opération de raffinement remplace chaque règle du système d'origine par un ensemble infini de règles, et nous avons de plus vu à la section 11.1 précédente qu'elle fait également disparaître certaines informations de descendance.

Aussi, pour raccourcir le chemin à parcourir entre un *CRS* faible et sa version avec partage, et pour faire en sorte que tout partage de deux sous-termes soit justifié par l'existence d'un ancêtre commun, nous allons présenter dans ce chapitre des étiquetages directs, qui se passent de raffinement et utilisent une et une seule règle étiquetée pour chaque règle du système d'origine. En contrepartie bien sûr, les mécanismes agissant sur les étiquettes seront définis à l'extérieur du formalisme des *CRS*.

L'étiquetage obtenu permettra pour certains systèmes de partager plus que l'étiquetage pleinement paresseux du chapitre 4. Cependant les deux étiquetages spécifieront le même partage sur le λ -calcul. Ainsi l'étiquetage défini à cette section est une alternative à l'étiquetage du chapitre 4 qui est encore une généralisation de la technique de Wadsworth.

Les étiquettes sont similaires à celles utilisées entre autres au chapitre 4.

Définition 11.4 (Étiquettes). *Soit L un ensemble dénombrable dont les éléments sont appelés **étiquettes initiales**. L'ensemble d'étiquettes \mathcal{L} est défini par la grammaire suivante :*

$$\alpha ::= p \mid [\alpha, \alpha] \quad p \text{ position}$$

Sur toute signature de *CRS* nous pouvons définir une notion de méta-terme étiqueté qui enrichit les méta-termes *CRS*.

Définition 11.5 (Termes étiquetés). *Soit une signature \mathcal{S} . Les **méta-termes étiquetés** sur \mathcal{S} sont définis par la grammaire suivante :*

$$t ::= x \mid \langle x \rangle^\alpha t \mid f^\alpha(t_1, \dots, t_n) \mid Z(t_1, \dots, t_n) \mid Z^\alpha(t_1, \dots, t_n)$$

Les notions de variables libres et liées (définition 0.61) sont immédiatement étendues à cette grammaire étiquetée.

Rappelons-nous que les méta-variables, dont la grammaire montre qu'elle peuvent ou non porter des étiquettes, ne servent qu'à exprimer les règles de réduction. En l'occurrence les méta-variables non étiquetées seront utilisées pour les membres gauches des règles tandis que les méta-variables étiquetées seront utilisées pour les membres droits des règles.

La notion d'étiquetage initial de la définition 4.39 et le réétiquetage uniforme de la définition 4.41 sont étendus de manière à donner des étiquettes aux méta-variables.

Définition 11.6 (Étiquetage initial). *Soit \mathcal{S} une signature CRS. Pour toute position p la fonction $i_p(\cdot)$ transforme un méta-terme sur \mathcal{S} en un méta-terme sur $\mathcal{S}^{\mathcal{L}}$ par les règles suivantes :*

$$\begin{aligned} i_p(x) &= x \\ i_p([x]t) &= [x]i_p(t) \\ i_p(f(t_1, \dots, t_n)) &= f^p(i_{p-1}(t_1), \dots, i_{p-n}(t_n)) \\ i_p(Z(t_1, \dots, t_n)) &= Z^p(i_{p-1}(t_1), \dots, i_{p-n}(t_n)) \end{aligned}$$

Soit t un méta-terme sur une signature \mathcal{S} . L'**étiquetage initial** de t est le méta-terme $i_\epsilon(t)$, que nous notons encore $i(t)$.

Définition 11.7 (Réétiquetage uniforme). *Pour tout terme t sur une signature $\mathcal{S}^{\mathcal{L}}$ et étiquette $\omega \in \mathcal{L}$, le réétiquetage uniforme $[\omega, t]$ est défini ainsi :*

$$\begin{aligned} [\omega, x] &= x \\ [\omega, [x]t] &= [x][\omega, t] \\ [\omega, f^\alpha(t_1, \dots, t_n)] &= f^{[\omega, \alpha]}([\omega, t_1], \dots, [\omega, t_n]) \\ [\omega, Z^\alpha(t_1, \dots, t_n)] &= Z^{[\omega, \alpha]}([\omega, t_1], \dots, [\omega, t_n]) \end{aligned}$$

Similairement à ce qui été fait à la définition 4.42, un étiquetage d'une règle est obtenu en prenant un étiquetage quelconque de son membre gauche, car tout terme étiqueté correspondant au motif doit pouvoir être réduit, et un étiquetage initial de son membre droit qui servira de base fixe à la construction des étiquettes définitives du réduit. À cela nous ajoutons une condition technique : les méta-variables du membre gauche ne sont pas étiquetées (alors que celles du membre droit le sont par définition de l'étiquetage initial). Ceci permettra de simplifier la définition des valuations étiquetées (définition 11.13).

Définition 11.8 (Règle étiquetée). *Soit $\kappa : l \rightarrow r$ une règle CRS. Notons L_α l'ensemble des étiquetages de l dont les méta-variables ne sont pas étiquetées. L'ensemble des **étiquetages de la règle** κ est l'ensemble de règles étiquetées*

$$\kappa^{\mathcal{L}} = \{ l_\alpha \rightarrow [\tau(l_\alpha), i(r)] \mid l_\alpha \in L_\alpha \}$$

Exemple 11.9.

Considérons la règle suivante tirée de l'exemple 11.1

$$\kappa_d : @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0))$$

Un étiquetage de la règle κ_d est la règle étiquetée

$$\kappa_d^l : @^\omega(\langle x \rangle^\alpha Z(x), Z_0) \rightarrow Z^{[\omega, \epsilon]}(Z^{[\omega, 1]}(Z_0^{[\omega, 11]}))$$

À partir de maintenant nous ne pouvons plus réutiliser directement la notion de réduction des CRS, car les modifications d'étiquettes doivent être répercutées aux positions dynamiques du réduit, c'est-à-dire à certaines des positions apportées par les valuations. Il nous faut donc définir des variantes étiquetées des notions originales de substituts (définition 0.66) et de valuations (définition 0.68).

La définition des substituts est immédiatement transcrite dans le cadre étiqueté.

Définition 11.10 (Substitut étiqueté). *Un substitut étiqueté n -aire est une fonction notée $\underline{\lambda}x_1 \dots x_n.t$ avec t un terme étiqueté, qui prend n termes étiquetés en entrée et renvoie le terme étiqueté défini par*

$$(\underline{\lambda}x_1 \dots x_n.t)(t_1, \dots, t_n) = t^{x_1 := t_1, \dots, x_n := t_n}$$

où la substitution est directement étendue de la définition 0.63.

La notion de valuation en revanche doit être légèrement étendue pour prendre en charge une certaine opération de réétiquetage quand cela est nécessaire, en fonction des éventuelles étiquettes que portent les méta-variables. Une valuation étiquetée a deux comportements : appliquée à une méta-variable non étiquetée, une valuation étiquetée a le même effet qu'une valuation « ordinaire » (c'est-à-dire selon la définition 0.68) ; en revanche, appliquée à une méta-variable étiquetée, une valuation étiquetée opère d'abord un réétiquetage des positions dynamiques du substitut correspondant.

Pour formaliser ce comportement, étendons la notion de réétiquetage uniforme (définition 4.41) aux substituts de manière à en réétiqueter exactement les positions dynamiques. Rappelons que par définition 0.66, les positions dynamiques d'un substitut $\underline{\lambda}x_1 \dots x_n.t$ sont les positions de t qui ne sont pas dans une expression libre de $\langle x_1 \dots x_n \rangle t$, ce qui signifie que les positions dynamiques de $\underline{\lambda}x_1 \dots x_n.t$ sont les positions du $\{x_1 \dots x_n\}$ -squelette de t .

Définition 11.11 (Réétiquetage d'un substitut). *Soit $\underline{\lambda}x_1 \dots x_n.t$ un substitut étiqueté et α une étiquette. Notons $s = \langle\langle t \rangle\rangle^{\{x_1, \dots, x_n\}}$ et $t = s[t_1, \dots, t_k]$. Alors le **ré-étiquetage** $[\alpha, \underline{\lambda}x_1 \dots x_n.t]$ du substitut $\underline{\lambda}x_1 \dots x_n.t$ par l'étiquette α est le substitut étiqueté défini par :*

$$[\alpha, \underline{\lambda}x_1 \dots x_n.t] = \underline{\lambda}x_1 \dots x_n.[\alpha, s][t_1, \dots, t_k]$$

Exemple 11.12.

Les éventuelles expressions libres d'un substitut ne sont pas réétiquetées, et en particulier le réétiquetage d'un substitut associé à une méta-variable d'arité nulle est sans effet.

$$\begin{aligned} [[\omega, 1], \underline{\lambda}x_1.f^\beta(x_1, x_1)] &= \underline{\lambda}x_1.f^{[[\omega, 1], \beta]}(x_1, x_1) \\ [[\omega, 1], \underline{\lambda}x_1.f^\beta(x_1, f^\delta(g^\zeta(a^\eta), x_1))] &= \underline{\lambda}x_1.f^{[[\omega, 1], \beta]}(x_1, f^{[[\omega, 1], \delta]}(g^\zeta(a^\eta), x_1)) \\ [[\omega, 2], \underline{\lambda}\phi.a^\gamma] &= \underline{\lambda}\phi.a^\gamma \end{aligned}$$

Nous pouvons donc étendre la définition 0.68 des valuations en ajoutant un cas pour les méta-variables étiquetées.

Définition 11.13 (Valuation étiquetée). Une **valuation étiquetée** σ assigne des substitués étiquetés n -aires aux méta-variables n -aires. Une valuation étiquetée sûre s'applique comme dans les CRS, mais introduit en plus une opération de réétiquetage quand elle rencontre une méta-variable étiquetée.

$$\begin{aligned} x^\sigma &= x \\ (\langle x \rangle^\alpha t)^\sigma &= \langle x \rangle^\alpha (t^\sigma) \\ (f^\alpha(t_1, \dots, t_n))^\sigma &= f^\alpha(t_1^\sigma, \dots, t_n^\sigma) \\ (Z(t_1, \dots, t_n))^\sigma &= (\sigma(Z))(t_1^\sigma, \dots, t_n^\sigma) \\ (Z^\alpha(t_1, \dots, t_n))^\sigma &= [\alpha, \sigma(Z)](t_1^\sigma, \dots, t_n^\sigma) \end{aligned}$$

Exemple 11.14.

Considérons la valuation étiquetée

$$\sigma = \{ Z \mapsto \underline{\lambda}x_1.f^\beta(x_1, x_1), Z_0 \mapsto \underline{\lambda}\phi.a^\gamma \}$$

Alors nous avons

$$\begin{aligned} (@^\omega(\langle x \rangle^\alpha Z(x), Z_0))^\sigma &= @^\omega(\langle x \rangle^\alpha f^\beta(x, x), a^\gamma) \\ (Z^{[\omega, \epsilon]}(Z^{[\omega, 1]}(Z_0^{[\omega, 11]})))^\sigma &= f^{[[\omega, \epsilon], \beta]}(f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma), f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma)) \end{aligned}$$

Remarque 11.15. Le réétiquetage d'un substitut de la forme $\underline{\lambda}\phi.t$ ne lui apporte aucune modification : $[\alpha, \underline{\lambda}\phi.t] = \underline{\lambda}\phi.[\alpha, \square][t] = \underline{\lambda}\phi.\square[t] = \underline{\lambda}\phi.t$. Par conséquent l'application d'une valuation étiquetée à une méta-variables étiquetée Z^α d'arité 0 peut ignorer l'étiquette de cette dernière.

Avec cette extension de la notion de valuation, la définition 0.75 d'un pas de réduction CRS peut être reproduite.

Définition 11.16 (Réduction étiquetée). Un **pas de réduction étiqueté** ρ est l'application d'une variante $\kappa : l \rightarrow r$ d'une règle étiquetée κ_0 avec une valuation étiquetée σ sûre pour κ , dans un contexte c . Un tel pas de réduction est noté $\rho : c[l^\sigma] \rightarrow c[r^\sigma]$ et nous appelons l^σ un **radical** et r^σ son **réduit**.

Exemple 11.17.

Considérons la règle κ_d tirée de l'exemple 11.1

$$\kappa_d : @(\langle x \rangle Z(x), Z_0) \rightarrow Z(Z(Z_0))$$

ainsi que le terme source étiqueté de l'exemple 11.2

$$@^\omega(\langle x \rangle^\alpha f^\beta(x, x), a^\gamma)$$

et l'étiquetage κ_d^l de la règle κ_d proposé à l'exemple 11.9

$$\kappa_d^l : @^\omega(\langle x \rangle^\alpha Z(x), Z_0) \rightarrow Z^{[\omega, \epsilon]}(Z^{[\omega, 1]}(Z_0^{[\omega, 11]}))$$

Alors nous avons un pas de réduction

$$@^\omega(\langle x \rangle^\alpha f^\beta(x, x), a^\gamma) \rightarrow f^{[[\omega, \epsilon], \beta]}(f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma), f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma))$$

dans la cible duquel les deux copies du sous-terme $f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma)$ sont partagées.

Cette nouvelle notion de réduction étiquetée, dont nous vérifierons à la section 11.3 suivante qu'elle définit un *SLS*, partage donc plus que l'étiquetage pleinement paresseux du chapitre 4. Nous pourrions également vérifier que dans le cadre du λ -calcul cet étiquetage est équivalent à celui du chapitre 2, c'est-à-dire qu'il spécifie le même partage que la technique de réduction de Wadsworth. Ainsi, nous venons de définir une généralisation aux *CRS* du partage pleinement paresseux de Wadsworth qui n'est pas équivalente à celle du chapitre 4, ni aux familles faibles définies à partir de l'étiquetage du chapitre 7.

11.3 Partage en réduction non-faible

Nous avons jusqu'ici travaillé avec des systèmes en réduction faible, et avons pu constater que dans ce cadre restreint qu'une certaine généralisation de la réduction pleinement paresseuse de Wadsworth (celle du chapitre 4) était une manière d'y interpréter l'optimalité à la Lévy. Cependant, il est assez évident que la réduction non-faible permet d'obtenir de meilleures stratégies. Nous pouvons par exemple constater que la réduction pleinement paresseuse n'est pas la réduction optimale à la Lévy pour le λ -calcul non-faible usuel, ou encore que le partage pleinement paresseux, même optimisé, autorise la duplication de pré-radicaux gelés, pré-radicaux qu'il pourrait donc être avantageux de réduire avant ces duplications.

Dans la présente section nous allons voir que l'étiquetage de la section 11.2 précédente définit un système de partage étiqueté *SLS*, et ce même en autorisant des réductions non-faibles. Cela nous permettra de considérer des stratégies de réduction alliant partage pleinement paresseux avec réduction non-faible, qui sont susceptibles de produire des séquences de réduction plus courtes que les réductions pleinement paresseuses faibles. Ceci sera illustré à la section 11.4.

La différence majeure dans la vérification des axiomes des *SLS* par rapport au chapitre 4 va concerner l'axiome *Partage* (définition 1.15), qui contrôle les cas dans lesquels deux positions de la zone d'effet d'un pas de réduction peuvent avoir la même étiquette. Dans la preuve du lemme 4.48 la vérification de cet axiome était triviale, car dans l'étiquetage du chapitre 4 les étiquettes d'une zone d'effet sont toutes distinctes. Ici, cette dernière propriété n'est plus vérifiée : dans la cible

$$f^{[[\omega, \epsilon], \beta]}(f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma), f^{[[\omega, 1], \beta]}(a^\gamma, a^\gamma))$$

du pas de réduction de l'exemple 11.17, la zone d'effet $\{\epsilon, 1, 2\}$ contient deux occurrences de la même étiquette $[[\omega, 1], \beta]$. Nous allons donc devoir effectuer une vérification un peu plus étoffée, de même qu'il avait été fait au chapitre 3 dans le cadre des β -systèmes parcimonieux (lemme 3.10) et des β -systèmes généreux (lemme 3.17) à l'aide de propriétés comme le respect local de la convention de Barendregt (définition 2.12).

La nouvelle stratégie de vérification sera suggérée par un autre aspect du système actuel : nous autorisons maintenant les réductions non-faibles. Ainsi, les remarques sur la non-nécessité des renommages de variables en réduction faible [FMS05] ne tiennent plus, et il va devenir nécessaire d'autoriser certains renommages si nous voulons que le système soit toujours capable d'exprimer toutes les séquences de réduction. Nous allons toujours considérer des termes syntaxiques, qui ne sont pas égalisés module α -conversion, mais maintenant les pas de réduction pourront faire tous les renommages de variables jugés utiles. En particulier, nous pouvons imposer que tout pas de réduction prenne en charge dans sa cible les renommages de variables nécessaires à la préservation de la convention de Barendregt, ainsi que cela a été évoqué dans la discussion de la section 3.5. Par conséquent, la rupture de la convention de Barendregt (définition 0.41) avec laquelle nous avons composé jusque-là (voir exemple 2.11) n'est plus une fatalité.

Le point crucial dans cette approche est de s'assurer que tous les renommages de variables ont bien lieu à l'intérieur de la zone d'effet du pas de réduction concerné, c'est-à-dire dans la région dont les étiquettes sont modifiées. Ou, en regardant dans l'autre sens, la zone d'effet doit toujours être assez grande pour contenir toutes les positions affectées par un renommage de variables. C'est justement le cas des zones d'effets construites sur des squelettes que nous utilisons ici. En l'occurrence, les renommages seront contenus dans les positions dynamiques des substituts, qui sont effectivement affectées par les réétiquetages de substituts (définition 11.11).

Définition 11.18 (Renommage d'un substitut). *Soient $\underline{\lambda}x_1\dots x_n.t$ et $\underline{\lambda}x_1\dots x_n.t'$ deux substituts étiquetés. Notons $s = \langle\langle t \rangle\rangle^{\{x_1, \dots, x_n\}}$, $s' = \langle\langle t' \rangle\rangle^{\{x_1, \dots, x_n\}}$, $t = s[t_1, \dots, t_n]$ et $t' = s'[t'_1, \dots, t'_n]$. Nous notons $\underline{\lambda}x_1\dots x_n.t =_\alpha \underline{\lambda}x_1\dots x_n.t'$ si :*

- pour tout i , $t_i = t'_i$, et
- $s =_\alpha s'$.

Nous notons $\sigma =_\alpha \sigma'$ si les valuations σ et σ' ont même domaine et si pour toute méta-variable Z de ce domaine $\sigma(Z) =_\alpha \sigma'(Z)$.

Définition 11.19 (Pas de réduction étiqueté avec renommage). *Un pas de réduction étiqueté avec renommage* ρ est l'application dans un contexte c d'une variante $\kappa : l \rightarrow r$ d'une règle étiquetée κ_0 avec une valuation σ sûre pour l et une valuation σ' sûre pour r et telles que $\sigma =_\alpha \sigma'$. Un tel pas de réduction est noté $\rho : c[l^\sigma] \rightarrow c[r^{\sigma'}]$.

Le pas ρ est un **pas de réduction étiqueté avec renommage partagé** si de plus $\mathbb{S}(l^\sigma)$ implique $\mathbb{S}(r^{\sigma'})$.

La condition de renommage partagé est là pour assurer que le renommage ne va pas rompre le partage, c'est-à-dire que des abstractions portant la même étiquette seront renommées de façon cohérente.

Nous pouvons maintenant adapter la définition d'un SLS et la vérification des axiomes de celles déjà vues à la section 4.3, mais avec une nouvelle vérification de l'axiome de partage.

Instanciation 11.20. Soit un CRS $(\mathcal{S}, \mathcal{R})$. Considérons la signature $\mathcal{S}^\mathcal{L} \cup \mathcal{X}$ où les variables sont considérées comme des symboles nullaires et définissons :

- \mathcal{T} est l'ensemble des termes sur $\mathcal{S}^\mathcal{L} \cup \mathcal{X}$ respectant la convention de Barendregt.
- \mathcal{R} est l'ensemble des pas de réduction étiquetés avec renommage partagé dont la source et la cible respectent la convention de Barendregt.

Soit $\rho \in \mathcal{R}$. Notons $\rho : c[l^\sigma] \rightarrow c[[\tau(l), r]^{\sigma'}]$ et définissons :

- $\text{dex}(\rho) = l^\sigma$
- $\text{duct}(\rho) = [\tau(l), r]^{\sigma'}$
- $\text{ctx}(\rho) = c$
- $\text{effz}(\rho) = \mathcal{P}_p(r, \sigma') \cup \mathcal{P}_d(r, \sigma')$

Notons $\Sigma = (\mathcal{S}^\mathcal{L} \cup \mathcal{X}, \mathcal{T}, \mathcal{R}, \text{dex}, \text{duct}, \text{ctx}, \text{effz})$.

Lemme 11.21 (ATRS marqué). *Le système Σ est un ATRS marqué.*

Démonstration.

Source & Cible Par définition toute source et toute cible de pas de réduction respecte la convention de Barendregt.

Résidus Identique au lemme 4.47.

Zone d'effet Identique au lemme 4.47. □

Lemme 11.22 (SLS). *Le triplet $(\mathcal{L}, \leq, \Sigma)$ est un SLS.*

Démonstration. Les axiomes *Étiquette racine*, *Progression* et *Héritage* se vérifient comme dans le lemme 4.48. Reste l'axiome *Partage*.

Partage Soit un pas de réduction $\rho : c[l^\sigma] \rightarrow c[[\tau(l), r]^{\sigma'}]$ tel que $\mathbb{S}(l^\sigma)$. Soient $p, q \in \text{effz}(\rho)$ tels que $\tau_p([\tau(l), r]^{\sigma'}) = \tau_q([\tau(l), r]^{\sigma'})$. Notons $\alpha = \tau_p([\tau(l), r]^{\sigma'}) = \tau_q([\tau(l), r]^{\sigma'})$. Par définition de la zone d'effet, p et q sont soit des positions propres soit des positions dynamiques de $[\tau(l), r]^{\sigma'}$. Trois cas sont à considérer :

- Si p et q sont deux positions propres de $[\tau(l), r]^{\sigma'}$, alors $\alpha = [\tau(l), p_0]$ et $\alpha = [\tau(l), q_0]$ avec $p_0 = q_0$ une position de r dont résultent p et q . Comme $p_0 = q_0$ nous avons $r|_{p_0} = r|_{q_0}$. Nous en déduisons $[\tau(l), r|_{p_0}]^{\sigma'} = [\tau(l), r|_{q_0}]^{\sigma'}$ et donc $[\tau(l), r]^{\sigma'}|_p = [\tau(l), r]^{\sigma'}|_q$.
- Si p ou q est une positions propre de $[\tau(l), r]^{\sigma'}$ et l'autre en est une position dynamique, alors α est à la fois de la forme $[\tau(l), \beta]$ et de la forme $[[\tau(l), \gamma], \delta]$, ce qui implique l'équation impossible $\tau(l) = [\tau(l), \gamma]$.
- Si p et q sont deux positions dynamiques de $[\tau(l), r]^{\sigma}$, alors $\alpha = [[\tau(l), p_0], \beta_p]$ et $\alpha = [[\tau(l), q_0], \beta_q]$ avec en particulier $\beta_p = \beta_q$, et où $p_0 = q_0$ est la position d'une méta-variable Z de r . Notons donc $r|_{p_0} = r|_{q_0} = Z^\gamma(t_1, \dots, t_n)$, $\sigma(Z) = \underline{\lambda}x_1 \dots x_n.t$, et $\sigma'(Z) = \underline{\lambda}x_1 \dots x_n.t'$. Alors les étiquettes (égales) β_p et β_q viennent respectivement de deux positions p_σ et q_σ du corps t du substitut $\sigma(Z)$. De l'hypothèse $\mathbb{S}(l^\sigma)$ nous déduisons que $t|_{p_\sigma} = t|_{q_\sigma}$ et comme $\sigma =_\alpha \sigma'$ nous avons de même $t'|_{p_\sigma} = t'|_{q_\sigma}$.

Nous allons de plus vérifier que les sous-termes $t'|_{p_\sigma}$ et $t'|_{q_\sigma}$ sont affectés de la même manière par les réétiquetages et les substitutions. En effet, soit une occurrence d'une variable x libre dans $t'|_{p_\sigma} = t'|_{q_\sigma}$, à position p_x . Alors par convention de Barendregt l'occurrence de x à la position $p_\sigma \cdot p_x$ dans t' est libre dans t' si et seulement si l'occurrence de x à la position $q_\sigma \cdot p_x$ dans t' est libre dans t' . Ainsi, une position $p_\sigma \cdot p'_\sigma$ est une position dynamique de $\underline{\lambda}x_1 \dots x_n.t'$ si et seulement si la position $q_\sigma \cdot p'_\sigma$ est une position dynamique de $\underline{\lambda}x_1 \dots x_n.t'$. Donc

$$[[\tau(l), p_0]\sigma'(Z)](t_1, \dots, t_n)|_{p_\sigma} = [[\tau(l), q_0]\sigma'(Z)](t_1, \dots, t_n)|_{q_\sigma}$$

et enfin

$$[\tau(l), r]^{\sigma'}|_p = [\tau(l), r]^{\sigma'}|_q$$

□

11.4 Réduction (non-)faible, (non-)orthogonalité, et optimisations

Nous allons voir dans cette section que la réduction non-faible peut être vue à la fois comme une source d'optimisations et comme une source de non-orthogonalité, et que les deux aspects sont liés. Pour simplifier la discussion, nous considérerons ici le λ -calcul, muni de l'étiquetage du chapitre 2 (qui est équivalent à l'instanciation sur le λ -calcul de l'étiquetage de la section 11.2).

Le λ -calcul est un système orthogonal, qui est donc en particulier confluent. De plus, par la section 10.3 nous savons que la β -réduction faible des λ -graphes est fortement confluite. Cependant, comme le montre l'exemple 11.23 suivant la β -réduction non-faible des λ -graphes n'est pas confluite. Remarquons que ce point n'apparaît pas dans les travaux de Wadsworth [Wad71], car ce dernier considère une stratégie de réduction particulière.

Exemple 11.23.

Considérons le λ -terme

$$t = (\lambda x.a((\lambda y.b)x))c(\lambda x.a((\lambda y.b)x))$$

où a , b , et c désignent trois variables libres, et considérons l'étiquetage suivant dans lequel les deux occurrences du sous-terme $\lambda x.a((\lambda y.b)x)$ sont partagées par leur étiquette commune γ :

$$t^l = @^\alpha(@^\beta(\lambda^\gamma x.@^\delta(a, @^\zeta(\lambda^\eta y.b, x)), c), \lambda^\gamma x.@^\delta(a, @^\zeta(\lambda^\eta y.b, x)))$$

Deux étapes de réduction parallèle étiquetée sont possibles au niveau des étiquettes β et ζ . Commençons par réduire en parallèle les deux radicaux $@^\zeta(\lambda^\eta y.b, x)$. Le squelette de $\lambda^\eta y.b$ étant dégénéré aucune étiquette n'est modifiée et nous obtenons le terme étiqueté

$$t_1 = @^\alpha(@^\beta(\lambda^\gamma x.@^\delta(a, b), c), \lambda^\gamma x.@^\delta(a, b))$$

Réduisons maintenant le radical $@^\beta(\lambda^\gamma x.@^\delta(a, b), c)$. Encore une fois aucune étiquette n'est modifiée car le squelette de $\lambda^\gamma x.@^\delta(a, b)$ est dégénéré et nous obtenons la forme normale

$$t_2 = @^\alpha(@^\delta(a, b), \lambda^\gamma x.@^\delta(a, b))$$

dans laquelle nous pouvons observer en particulier que les deux occurrences du sous-terme $@(a, b)$ sont partagées via l'étiquette δ .

Repartons de t^l et réduisons directement le radical d'étiquette β . Cette fois le squelette de la λ -abstraction $\lambda^\gamma x.@^\delta(a, @^\zeta(\lambda^\eta y.b, x))$ est $\lambda^\gamma x.@^\delta(\square, @^\zeta(\square, x))$, ce qui amène quelques modifications d'étiquettes et le résultat suivant :

$$t_3 = @^\alpha(@^{[\beta, \delta]}(a, @^{[\beta, \zeta]}(\lambda^\eta y.b, c)), \lambda^\gamma x.@^\delta(a, @^\zeta(\lambda^\eta y.b, x)))$$

Dans ce terme, les deux radicaux anciennement partagés par l'étiquette ζ sont maintenant séparés, l'un conservant l'étiquette ζ et l'autre ayant changé pour $[\beta, \zeta]$. Il faut donc maintenant deux étapes de réduction partagée pour obtenir la forme normale :

$$t_4 = @^\alpha(@^{[\beta, \delta]}(a, b), \lambda^\gamma x.@^\delta(a, b))$$

qui n'est pas égale à la forme normale t_2 donnée par l'autre séquence de réduction. En particulier les deux occurrences du sous-terme $@(a, b)$ sont maintenant séparées, l'une marquée par l'étiquette δ et l'autre par $[\beta, \delta]$.

Ainsi, différentes séquences de réduction de graphes à partir d'un même λ -terme t peuvent aboutir à des graphes \mathcal{G}_2 et \mathcal{G}_4 différents. Ces deux graphes \mathcal{G}_2 et \mathcal{G}_4 représentent évidemment le même λ -terme t' (le λ -calcul est tout de même confluent), mais ils ne réalisent pas le même partage sur les sous-termes de t' .

Dans l'exemple 11.23 cette rupture de la confluence est basée sur l'instabilité des squelettes par réduction non-faible, c'est-à-dire la négation du lemme 2.8. Nous avons à l'origine une λ -abstraction $\lambda x.a((\lambda y.b)x)$ dont le squelette est

$\lambda x. \square(\square x)$. Cependant, la réduction de $(\lambda y. b)x$ en b supprime l'unique occurrence de la variable liée x , trivialisant au passage le squelette de la λ -abstraction externe λx . Remarquons que cette réduction serait interdite dans le λ -calcul faible, du fait justement de la présence de la variable x .

Ainsi, dans les systèmes faibles les squelettes sont des structures qui ne sont pas affectées par la réduction justement parce que la réduction faible empêche toute réduction susceptible de les corrompre. Ceci avait permis d'aller aux chapitres 8 et 9 jusqu'à résumer chaque squelette par un symbole atomique. En réduction non-faible en revanche les squelettes peuvent être transformés de nombreuses manières et ne peuvent plus être considérés comme des briques élémentaires de la réécriture.

Ce défaut de confluence causé par une réduction à l'intérieur d'un squelette met le doigt sur un certain défaut d'orthogonalité de la réduction non-faible. Ce fait était étalé sous nos yeux depuis le chapitre 4, où nous avons défini le raffinement canonique d'un système (définition dans le corollaire 4.31). Nous y avons vu en effet que pour exprimer la réduction étiquetée (c'est-à-dire la réduction de graphes) sans sortir des cadres de réécriture usuels il fallait intégrer aux règles de réduction les squelettes des abstractions contenues par les règles. Par conséquent, toute possibilité de réduction à l'intérieur d'un squelette fait apparaître une paire critique dans ces règles étendues et détruit l'orthogonalité du système pour nous ramener à la notion d'« orthogonalité en réduction faible » (définition 4.33 et lemme 4.34) qui a moins de sens en réduction non-faible.

Regardons maintenant ce même phénomène, mais du point de vue du partage. Le défaut de confluence sur les graphes signifie que différentes stratégies de réduction vont générer des partages différents du même terme. En l'occurrence la séquence de $t^l \Rightarrow t_1 \Rightarrow t_2$ partage plus que la séquence $t^l \Rightarrow t_3 \Rightarrow t_4$. C'est visible dans les termes étiquetés t_2 et t_4 , mais aussi dans les étapes de réduction : il faut deux étapes de réduction distinctes (d'étiquettes $[\beta, \zeta]$ et ζ) pour réduire depuis t_3 les résidus des deux radicaux $@^\zeta(\lambda^n y. b, x)$ qui étaient partagés dans t^l . Cette séparation est due à l'étape de réduction $t^l \Rightarrow t_3$, dont le radical recouvre l'un des radicaux de l'étape $t^l \Rightarrow t_1$.

Ainsi, un recouvrement implique ici des duplications. Plus précisément, tout pas de réduction $(\lambda x. t)u$ entraînant un réétiquetage (c'est-à-dire une copie) du squelette s de son abstraction $\lambda x. t$, un recouvrement au niveau de ce squelette s implique une copie du radical (nécessairement non-faible) dont la racine est dans ce squelette s . Dans un tel cadre, nous évitons une duplication en réduisant d'abord le radical non-faible, qui est aussi le plus profond.

Nous pouvons alors obtenir de meilleurs stratégies de réduction en combinant le partage pleinement paresseux avec les idées suivantes pour la sélection des radicaux à réduire :

1. Sélectionner un radical nécessaire $r_1 = (\lambda x. t)u$.
2. En cas de recouvrement, c'est-à-dire si le squelette de $\lambda x. t$ contient la racine d'un radical r_2 , et si de plus la nécessité de r_2 est établie, alors sélectionner r_2 et reprendre cette phase.

3. En cas d'absence de recouvrement, ou en cas de recouvrement avec un radical r_2 dont la nécessité n'est pas établie, réduire r_1 et les radicaux partagés avec r_1 .

Une telle stratégie simple, proposée par Thyer [Thy99], prend comme critère de mettre la λ -abstraction $\lambda x.t$ en forme normale de tête avant d'effectuer l'application à u . Une autre piste plus ambitieuse serait la reproduction dans ce cadre des mécanismes de la paresse dite « complète » proposée par Holst et Gomard [HG91], et définie initialement en termes d'évaluation pleinement paresseuse et d'évaluation partielle, avant d'être liée à des mécanismes de partage de contexte par Sinot [Sin08].

Finalement, du point de vue du partage la réduction non-faible est une cause de non-orthogonalité presque comme une autre, mais elle est aussi la possibilité d'éviter certaines duplications. Et finalement, la capacité des *SLS* à gérer les systèmes non-orthogonaux, qui pouvait être vue à l'origine comme un accessoire ne servant qu'à jouer avec des exemples bancals, apparaît maintenant comme une caractéristique cruciale pour la description de stratégies de réduction optimisées pour le simple λ -calcul.

11.5 Zones d'effet optimisées

La notion de squelette d'une abstraction a été la brique fondamentale et inébranlable qui a servi à bâtir les chapitres 2 à 10, en raison de sa stabilité en réduction faible. L'introduction de pas de réduction non-faibles fait s'effriter cet atome, et nous demande de revenir sur ce qui a été fait et dit dans cette thèse. Peut-être d'autres choses intéressantes ont-elles été cachées derrière ce masque et aspirent maintenant à retrouver la lumière.

Rappelons-nous le chapitre 3. Plusieurs définitions de la réduction pleinement paresseuse y ont été données pour le λ -calcul, et il est ressorti de l'analyse de ces définitions qu'elles proposaient trois notions différentes de partage. Ce résultat était bien embarrassant au moment où l'enjeu était de démontrer l'unité de la littérature sur le sujet de la pleine paresse, mais la restriction à la réduction faible nous a permis de conclure avec soulagement que ces différences n'avaient pas d'incidence sur la réduction.

En effet, les différences dans le partage ne pouvaient survenir qu'à des positions gelées, c'est-à-dire à des endroits où toute réduction était interdite. Nous en avons déduit en particulier que la duplication de l'épine d'une abstraction était équivalente à la duplication de tout son squelette et de manière plus générale qu'il ne servait à rien d'isoler un fragment particulier à l'intérieur d'un squelette.

Cette conclusion bien sûr ne tient plus en réduction générale, et il peut maintenant être intéressant de préférer un système construit sur les épines comme \mathcal{P}_{BLM} à un construit sur les squelettes comme \mathcal{P}_W . Dans un tel système, l'idée est de changer les étiquettes exactement des positions soumises à une substitution. Une manière générique de définir un tel étiquetage consiste à simplement modifier la définition 11.11 de réétiquetage d'un substitut $\underline{\lambda}x_1\dots x_n.t$:

au lieu d'appliquer le réétiquetage aux positions dynamiques de $\underline{\lambda}x_1\dots x_n.t$, qui sont définies par le $\{x_1, \dots, x_n\}$ -squelette $\langle\langle t \rangle\rangle^{\{x_1, \dots, x_n\}}$ de t , ne réétiqueter que la $\{x_1, \dots, x_n\}$ -épine $\langle t \rangle^{\{x_1, \dots, x_n\}}$ de t (définition 4.4).

Définition 11.24 (Réétiquetage d'un substitut). *Soit $\underline{\lambda}x_1\dots x_n.t$ un substitut étiqueté et α une étiquette. Notons $s = \langle t \rangle^{\{x_1, \dots, x_n\}}$ et $t = s[t_1, \dots, t_k]$. Alors le **réétiquetage** $[\alpha, \underline{\lambda}x_1\dots x_n.t]$ du substitut $\underline{\lambda}x_1\dots x_n.t$ par l'étiquette α est le substitut étiqueté défini par :*

$$[\alpha, \underline{\lambda}x_1\dots x_n.t] = \underline{\lambda}x_1\dots x_n.[\alpha, s][t_1, \dots, t_k]$$

Cette définition étant posée, nous pouvons refaire à l'identique les constructions des sections 11.2 et 11.3, à un détail près : les zones d'effets ne seront pas toujours assez grandes pour permettre les renommages de variables nécessaires au respect de la convention de Barendregt (voir section 3.5). Deux solutions sont : ajouter une règle explicite de renommage, à appliquer quand cela est nécessaire, ce qui sera fait dans l'exemple de la section 11.6 suivante, ou étendre les zones d'effet exactement aux positions affectées par un renommage. Cette dernière solution peut être vue comme une intégration de la première aux règles principales du système.

11.6 Un dernier exemple

Nous allons maintenant développer un exemple avec réduction non-faible et avec un réétiquetage guidé par les épines plutôt que les squelettes. Cet exemple est un étiquetage d'un calcul inspiré du λ -calcul en appel par nécessité [AFM⁺95], qui montre en quoi les SLS permettent de décrire des graphes un peu plus généraux que les graphes-termes d'ordre supérieur habituels [Kah98]. Cet exemple fournira une stratégie de réduction avec partage « plus-que-pleinement » paresseux.

Dans le λ -calcul en appel par nécessité λ_{let} , la règle (β) est remplacée par une règle d'introduction d'une construction `let ... in ...` qui reporte la substitution de l'argument.

$$(\lambda x.t)u \rightarrow \text{let } x = u \text{ in } t$$

Du point de vue de l'argument u , ceci permet de conserver une seule copie partagée de u pour un nombre potentiellement grand d'occurrences de x correspondantes dans t .

Cependant, dans notre contexte de SLS les étiquettes peuvent déjà prendre en charge ce partage. En particulier, la substitution de u ne modifie pas les étiquettes de u , qui reste donc partagé. Ce garde-fou nous permet d'abandonner un certain nombre de contraintes des règles de λ_{let} , et d'utiliser le report introduit par `let` d'une nouvelle manière. Changeons de point de vue. Du point de vue de t , dans l'expression

$$\text{let } x = u \text{ in } t$$

la construction `let` a pour effet d'abstraire la valeur de x . Ceci nous permet de partager plusieurs copies de t potentiellement liées par des `let` différents avec des arguments différents.

Définissons maintenant notre syntaxe et nos règles étiquetées pour ce système. Nous allons fabriquer le système étiqueté directement, afin d'alléger les notations. Il pourra être constaté que l'étiquetage général des *CRS* présenté à ce chapitre muni de la limitation des réétiquetages décrite à la section 11.5 aurait défini un étiquetage possédant essentiellement les mêmes propriétés.

Considérons l'ensemble \mathcal{L} des étiquettes formées par la grammaire

$$\alpha ::= p \mid [\alpha] \mid [\alpha, \alpha] \quad p \text{ position}$$

muni de l'ordre \leq généré par les conditions $\alpha \leq [\alpha]$ et $\alpha \leq [\alpha, \beta]$ pour toutes étiquettes $\alpha, \beta \in \mathcal{L}$. Définissons les termes étiquetés par la grammaire suivante, dans laquelle seules les applications et les `let` sont étiquetés. Étiqueter aussi les λ -abstractions serait possible, mais ce n'est pas nécessaire puisque le formalisme des *SLS* n'impose l'étiquetage que des racines des radicaux (axiome *Étiquette racine*, définition 1.15). Les *réponses* représentent les λ -abstractions sous des `let`. Les *contextes d'évaluation* servent à définir la stratégie de réduction en désignant l'un des radicaux à réduire en parallèle.

Termes (\mathcal{T}_{let}) :	$t ::= x \mid \lambda x.t \mid @^\alpha(t, t) \mid \text{let}^\alpha x = t \text{ in } t$	$x \in \mathcal{X}$
Réponses (\mathcal{A}_{let}) :	$A ::= \lambda x.t \mid \text{let}^\alpha x = t \text{ in } A$	$x \in \mathcal{X}$
Cont. éval. (\mathcal{E}_{let}) :	$E ::= _ \mid @^\alpha(E, t) \mid \text{let}^\alpha x = t \text{ in } E$	$x \in \mathcal{X}$

Les variables libres et liées sont définies comme d'habitude, en sachant que $\lambda x.t$ et `let` $x = u$ `in` t lient x dans t . Les actions de substitution et de réétiquetage sont combinées dans l'opération suivante : la *substitution étiquetée* $t^{x:=\omega, u}$ substitue u à x dans t et modifie les étiquettes de la x -épine de t en utilisant ω . Formellement :

$$t^{x:=\omega, u} = t \quad x \notin \text{fv}(t)$$

Sinon :

$$x^{x:=\omega, u} = u$$

$$(\lambda y.t)^{x:=\omega, u} = \lambda y.t^{x:=\omega, u} \quad y \notin \text{fv}(u)$$

$$(@^\alpha(t_1, t_2))^{x:=\omega, u} = @^{[\omega, \alpha]}(t_1^{x:=\omega, u}, t_2^{x:=\omega, u})$$

$$(\text{let}^\alpha y = v \text{ in } t)^{x:=\omega, u} = \text{let}^{[\omega, \alpha]} y = v^{x:=\omega, u} \text{ in } t^{x:=\omega, u} \quad y \notin \text{fv}(v)$$

Deux règles principales (Intro) et (Subst) découpent la β -réduction en deux phases :

- (Intro) introduit un `let` pour reporter la substitution d'un pas de β -réduction, et
- (Subst) effectue la substitution quand le contexte d'évaluation rencontre une variable.

Une règle additionnelle (Scope) empêche les `let` de cacher des β -radicaux. Une règle (GC) est ajoutée pour alléger les exemples. Enfin, une règle (Rename) de renommage explicite permet d'assurer que la réduction ne peut pas bloquer sur

des problèmes de capture de variables. Le système n'est pas orthogonal, car les règles (GC) et (Rename) génèrent un certain nombre de recouvrements. La réduction étiquetée est définie par les cinq règles :

Intro :	$@^\alpha(\lambda x.t, u) \rightarrow \text{let}^\alpha x = u \text{ in } t$	
Subst :	$\text{let}^\alpha x = t \text{ in } E[x] \rightarrow \text{let}^{[\alpha]} x = t \text{ in } E[y]^{y:=\alpha, t}$	y variable fraîche
Scope :	$@^\alpha(\text{let}^\beta x = t \text{ in } A, u) \rightarrow \text{let}^{[\alpha]\beta} x = t \text{ in } @^{[\alpha]}(A, u)$	$x \notin \text{fv}(u)$
GC :	$\text{let}^\alpha x = t \text{ in } u \rightarrow u$	$x \notin \text{fv}(u)$
Rename :	$\text{let}^\alpha x = t \text{ in } u \rightarrow \text{let}^{[\alpha]} y = t \text{ in } u^{x:=\alpha, y}$	$y \notin \text{fv}(u) \cup \text{bv}(u)$

Remarquons que dans la règle (Subst) une seule occurrence de la variable est substituée, et que les étiquettes sont modifiées le long du chemin de propagation de cette substitution.

Dans la séquence suivante nous suivons le sous-terme souligné $\underline{@^\delta(\lambda z.z, y)}$ et ses descendants. Dans le λ -calcul en appel par nécessité usuel, deux copies de $\underline{@^\delta(\lambda z.z, y)}$ sont séparées dès l'étape (1), quand la valeur de la première occurrence de x est requise. Même dans un cadre pleinement paresseux, seule la copie de $(\lambda z.z)$ est évitée, ce qui ne change rien ici. Dans ce système en revanche, les deux copies de $\underline{@^\delta(\lambda z.z, y)}$ restent partagées jusqu'à la fin de l'étape (5). Entre temps, les étapes (3), (4), et (5) évaluent en parallèle les deux copies. Par conséquent, la fonction substituée à x lors de l'étape (6) est déjà complètement évaluée. Le nouveau comportement ici est le partage de deux sous-terms apparaissant à des positions de natures très différentes : l'un est enterré dans le squelette d'une λ -abstraction tandis que l'autre est sous les projecteurs, dans le contexte d'évaluation.

$$\begin{aligned}
& @^\alpha(\lambda x.@^\beta(x, @^\gamma(x, a)), \lambda y.\underline{@^\delta(\lambda z.z, y)}) \\
\rightarrow_I & \text{let}^\alpha x = \lambda y.\underline{@^\delta(\lambda z.z, y)} \text{ in } @^\beta(x, @^\gamma(x, a)) \\
\rightarrow_{Su} & \text{let}^{[\alpha]} x = \lambda y.\underline{@^\delta(\lambda z.z, y)} \text{ in } @^{[\alpha, \beta]}(\lambda y.\underline{@^\delta(\lambda z.z, y)}, @^\gamma(x, a)) \tag{1} \\
\rightarrow_I & \text{let}^{[\alpha]} x = \lambda y.\underline{@^\delta(\lambda z.z, y)} \text{ in } \text{let}^{[\alpha, \beta]} y = @^\gamma(x, a) \text{ in } \underline{@^\delta(\lambda z.z, y)} \tag{2} \\
\Rightarrow_I & \text{let}^{[\alpha]} x = \lambda y.\underline{\text{let}^\delta z = y \text{ in } z} \text{ in } \text{let}^{[\alpha, \beta]} y = @^\gamma(x, a) \text{ in } \underline{\text{let}^\delta z = y \text{ in } z} \tag{3} \\
\Rightarrow_{Su} & \text{let}^{[\alpha]} x = \lambda y.\underline{\text{let}^\delta z = y \text{ in } \underline{y}} \text{ in } \text{let}^{[\alpha, \beta]} y = @^\gamma(x, a) \text{ in } \underline{\text{let}^\delta z = y \text{ in } \underline{y}} \tag{4} \\
\Rightarrow_{GC} & \text{let}^{[\alpha]} x = \lambda y.\underline{y} \text{ in } \text{let}^{[\alpha, \beta]} y = @^\gamma(x, a) \text{ in } \underline{y} \tag{5} \\
\rightarrow_{GC}^{Su} & \text{let}^{[\alpha]} x = \lambda y.\underline{y} \text{ in } @^\gamma(x, a) \\
\rightarrow_{GC}^{Su} & @^{[[\alpha], \gamma]}(\lambda y.\underline{y})a \tag{6}
\end{aligned}$$

Le terme étiqueté obtenu à l'étape (2) et le graphe correspondant sont représentés ci-après, et montrent comment les *SLS* peuvent dans certains cas admettre le partage de sous-terms ouverts affectés par des lieurs différents. Ce genre de graphe est appelé *non-admissible* par Wadsworth 2, mais dans notre cas le terme étiqueté montre qu'il n'y a pas d'ambiguïté dans les liaisons et que l'utilisation de tels graphes est donc envisageable. En revanche, ici le renommage de variables ne peut pas être libre. En effet, renommer l'une des occurrences de y en y' demande de rompre le partage et de modifier l'une des occurrences de l'étiquette δ . Ce phénomène ne peut survenir lorsque nous réétiquetons les

squelettes, mais apparaît ici avec le réétiquetage limité aux épines. Ainsi, si ces graphes « non-admissibles » peuvent être représentés dans le formalisme des *CRS*, leur manipulation peut dépendre de la manière dont sont représentées les variables.

$$\begin{array}{l}
 \text{let}^{[[\alpha]]} x = \lambda y. @^\delta(\lambda z.z, y) \\
 \text{in } \text{let}^{[[[\alpha]\beta]]} y = @^\gamma(x, a) \\
 \quad \text{in } @^\delta(\lambda z.z, y) \\
 \rightarrow \text{let}^{[[\alpha]]} x = \lambda y. @^\delta(\lambda z.z, y) \\
 \text{in } \text{let}^{[[[[\alpha]\beta]]]} y = @^\gamma(x, a) \\
 \quad \text{in } @^{[[[\alpha]\beta]]\delta}(\lambda z.z, y)
 \end{array}
 \quad \left| \quad
 \begin{array}{c}
 \begin{array}{c}
 \text{xin} \\
 \lambda y \quad \text{yin} \\
 \text{---} \\
 x \quad a \\
 \text{---} \\
 @ \\
 \text{---} \\
 \lambda z \quad y \\
 \text{---} \\
 z
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{c}
 \text{xin} \\
 \lambda y \quad \text{y'in} \\
 \text{---} \\
 x \quad a \\
 \text{---} \\
 @ \\
 \text{---} \\
 \lambda z \quad y' \\
 \text{---} \\
 z
 \end{array}
 \end{array}$$

Finalement, le formalisme de partage de sous-termes des *SLS* permet d'exprimer des graphes un peu plus généraux que les graphes-termes d'ordre supérieur usuels, et permet ainsi d'exprimer simplement des stratégies de réduction qui vont au-delà de l'appel par nécessité et de l'évaluation pleinement paresseuse.

Bilan du chapitre

Cet ultime chapitre a montré trois aspects des outils présentés dans cette thèse qui permettent d'aller au-delà de l'optimalité faible :

- Nous pouvons ajouter du partage à l'intérieur de la zone d'effet d'une règle, ce qui permet à de nouveaux nœuds du graphe d'être partagés dès leur création.
- Nous pouvons autoriser des réductions non-faibles, ce qui permet l'élaboration de stratégies de réduction optimisées.
- Nous pouvons diminuer l'étendue des zones d'effet, ce qui permet de limiter le nombre de nœuds créés par une étape de réduction.

Et tout ceci peut se faire uniquement avec du partage de sous-termes, et dans le formalisme simple des *systèmes de partage étiqueté SLS*.

Conclusion générale

En complément des conclusions partielles et des discussions réparties dans l'ensemble de ce document, ce chapitre conclusif évoque trois résultats, trois points de méthode, et trois pistes de travaux futurs auxquels je suis particulièrement attaché.

Résultats

Au fil de ce document nous avons généralisé au cadre de la réécriture du second ordre plusieurs concepts venus du λ -calcul, et les avons combinés d'une manière qui donne de nouveaux éclairages sur ces deux mondes à la fois.

Cela a concerné en premier lieu la technique d'évaluation du λ -calcul de Wadsworth [Wad71], qui limite le nombre de pas de β -réduction en associant deux mécanismes : une stratégie de réduction « par la gauche » qui évite les pas de calcul non-nécessaires, et une représentation des λ -termes sous forme de graphes qui permet d'exprimer le partage de certains sous-termes et ainsi de limiter la duplication des pas de calcul restant à effectuer. Nous avons dissocié ces deux mécanismes et proposé au chapitre 4 une notion de réduction partagée « pleinement paresseuse » applicable à l'ensemble des *CRS* et prouvablement équivalente à la réduction partagée de Wadsworth dans le cas particulier du λ -calcul (théorème 4.52 de partage équivalent).

Nous avons également porté à l'ensemble des *CRS* la notion de réduction faible combinatoire proposée par Howard [How70] et explicitée par Çağman et Hindley [cH98]. Cette notion de réduction restreinte n'a pas été utilisée dans cette thèse pour son objet initial, qui était la représentation dans la syntaxe du λ -calcul du comportement opérationnel de la logique combinatoire, mais parce qu'elle présente un comportement plus simple que la β -réduction générale tout en permettant néanmoins d'exprimer les stratégies de réduction des langages de programmation usuels.

Enfin, au chapitre 7 nous avons proposé une notion de famille de radicaux qui, dans le cadre des *CRS* orthogonaux restreints à la réduction faible, reproduit la notion de famille développée par Lévy pour le λ -calcul [Lé78, Lé80]. Ces familles en réduction faible représentent une partition des familles de Lévy et sont une instance de la notion abstraite que proposent Glauert et Khasidashvili [GK96] des familles de radicaux (théorème 10.40). Elles caractérisent ainsi une notion « d'optimalité faible ».

Ces constructions étant en place, nous avons démontré que la réduction partagée pleinement paresseuse correspondait à la réduction de familles faibles complètes (théorèmes 7.29 et 7.36), c'est-à-dire à la notion de réduction caractérisant l'optimalité faible. Nous avons ainsi établi que :

*L'évaluation pleinement paresseuse à la Wadsworth
est une réalisation de la réduction optimale à la Lévy
pour les systèmes de réécriture du second ordre orthogonaux faibles.*

La réduction partagée pleinement paresseuse/réduction optimale faible a de plus été liée à deux autres notions de réduction, qui nous en apprennent plus sur le comportement de la réduction faible et sur ce qu'apporte le partage dans ce cadre.

Nous avons d'abord au chapitre 9 établi un lien fort entre la réduction faible d'un CRS arbitraire et la réduction d'un système de réécriture du premier ordre. Ce lien est avant tout une bisimulation : à chaque pas de réduction dans l'un des deux mondes correspond exactement un pas dans l'autre, et les longueurs des séquences de réduction sont ainsi strictement préservées. Mais ce lien préserve en outre plusieurs autres aspects du comportement des systèmes, comme leur éventuelle orthogonalité, comme les notions de descendance, de résiduation, et de nécessité, ou encore comme le partage. Par conséquent cette bisimulation préserve enfin les familles de radicaux et l'optimalité. Cette accumulation de propriétés positives donne de solides états à la remarque :

*La réduction faible donne aux systèmes d'ordre supérieur
le comportement opérationnel de systèmes du premier ordre.*

Nous avons ainsi approfondi ce constat qui apparaissait déjà à plusieurs endroits dans la littérature², en particulier avec la possibilité de se passer du renommage des variables dans des systèmes en réduction faible avec substitutions explicites [FMS05], et avec la possibilité de compiler des programmes fonctionnels en appel par valeur ou appel par nécessité vers des machines abstraites ayant la forme de systèmes de réécriture du premier ordre [Ses97, ABDM03].

Nous avons ensuite au chapitre 10 montré que la réduction pleinement paresseuse dans les CRS faibles orthogonaux, à l'instar de la réduction partagée des systèmes du premier ordre, ne permettait pas d'obtenir des séquences de réduction strictement plus courtes que celles offertes par les stratégies de réduction faible sans partage : la réduction faible sans partage et la réduction faible avec partage pleinement paresseux sont soumises aux mêmes bornes inférieures. En revanche, en neutralisant les duplications la réduction partagée rend plus facile l'atteinte de cette borne. Alors que sans partage l'optimal ne peut être obtenu qu'en réduisant à chaque pas un radical nécessaire ne dupliquant aucun autre radical nécessaire, il suffit avec partage de réduire un radical nécessaire arbitraire. La différence de difficulté entre les deux entreprises devient particulièrement frappante en remarquant que dans des systèmes comme le λ -calcul faible les

2. et qui m'a été explicitement mis sous les yeux par Vincent van Oostrom

stratégies optimales sans partage ne sont pas calculables (théorème 10.6) alors qu'il existe au contraire des stratégies nécessaires partagées très simples, à commencer par celle de Wadsworth [Wad71]. Finalement :

*La réduction partagée n'améliore pas la borne inférieure
sur la longueur des séquences de réduction faible.
Mais elle la rend effectivement accessible.*

Méthodes

Ces principaux résultats ont été obtenus par un recours récurrent à deux structures, l'une adaptée au partage de sous-termes et l'autre caractéristique de la réduction faible combinatoire.

La première structure qui a constamment été mise en avant concerne la représentation du partage par des termes étiquetés. De même que la technique de Wadsworth [Wad71], mais à la différence des algorithmes réalisant l'optimalité à la Lévy en λ -calcul non-faible [Lam90, GAL92, AG98], toutes les notions de réduction partagée utilisées dans cette thèse sont mises en œuvre par des graphes opérant un partage de sous-termes. Ces graphes sont acycliques et ont une forme leur permettant d'être représentés par des termes étiquetés, où chaque étiquette représente un nœud d'un graphe. De plus, chaque étape de réduction de graphes peut être elle-même représentée par la réduction en parallèle d'un certain ensemble de radicaux de termes qui sont disjoints et identifiés par leurs étiquettes. Au chapitre 1 nous avons généralisé les techniques utilisées dans ce domaine par Maranget [Mar92, Mar91] (sur les systèmes du premier ordre orthogonaux) à un cadre abstrait, les systèmes de partage étiqueté *SLS*, permettant d'appliquer cette idée à tous les systèmes (d'ordre supérieur et pas toujours orthogonaux) rencontrés dans cette thèse.

*La réduction avec partage de sous-termes
peut être représentée par des séquences de réduction de termes étiquetés.*

Cette représentation sous forme de termes étiquetés a notamment rendu possible la comparaison de plusieurs systèmes de réduction de graphes apparentés au partage pleinement paresseux (théorème 3.26 de partage équivalent) et la démonstration de l'équivalence entre la réduction pleinement paresseuse et la réduction de familles faibles complètes (théorème 7.29). L'utilisation d'étiquettes nous a de plus donné accès à quelques astuces techniques, par exemple pour l'étude de la descendance (section 9.4).

La deuxième structure fondamentale, également omniprésente mais de manière peut-être moins constamment visible, est la notion de squelette d'une abstraction. Cette dernière caractérise en quelque sorte la « portée » d'une abstraction, et affleure en négatif dans la littérature quand est posée la question de « ce qui peut être extrait d'une abstraction ». Wadsworth en particulier caractérise des expressions qui peuvent être extraites d'une λ -abstraction $\lambda x.t$ (*abstractable expressions* en version originale [Wad71], puis rebaptisées « expressions libres » [PJ87]) et qui sont exactement les parties de $\lambda x.t$ qui ne sont pas

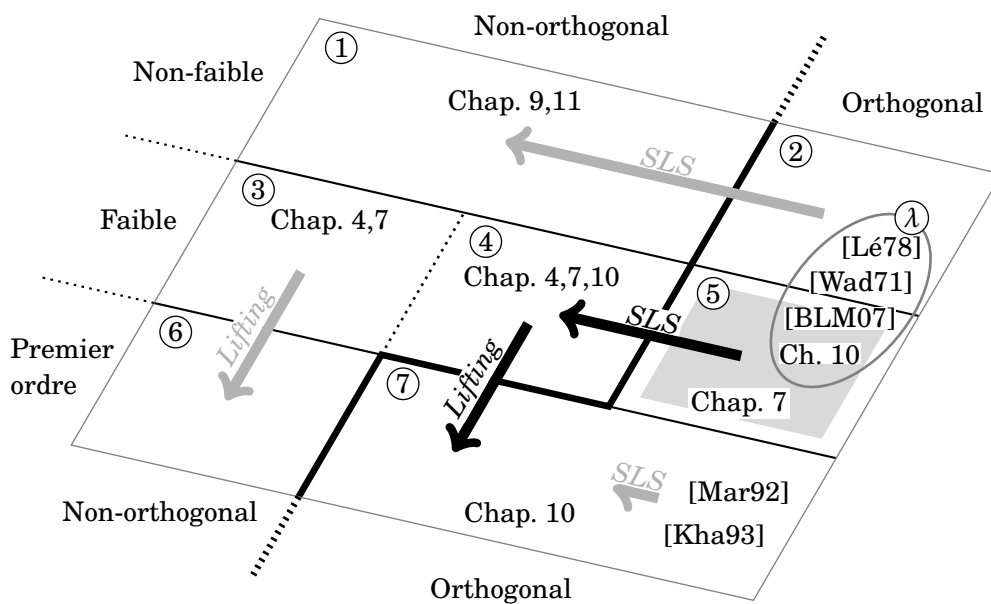
dupliquées lors d'une β -réduction utilisant cette λ -abstraction. La partie qui est dupliquée, c'est-à-dire le complément, est le squelette. La réduction faible combinatoire de Howard, Çağman et Hindley [How70, cH98] étant justement basée sur la portée des lieurs, nous avons pu observer dans cette thèse que les squelettes des abstractions sont des fragments de termes invariants en réduction faible, et peuvent donc être utilisés comme des éléments de structure et des guides du raisonnement.

*En réduction faible,
les squelettes des abstractions
sont les briques élémentaires structurant les termes.*

Ainsi, les squelettes ont d'abord servi de base aux étiquetages de tous les systèmes rencontrés. Ils ont également servi à définir la notion de raffinement canonique d'un CRS (corollaire 4.31), dont nous avons remarqué au chapitre 4 qu'il est à la fois le plus petit raffinement permettant d'éliminer les méta-variables d'arité non nulle et le plus grand raffinement préservant l'« orthogonalité en réduction faible », et au chapitre 9 qu'il est exactement le raffinement permettant d'exprimer toutes les opérations de *lifting* pouvant affecter un méta-terme.

La décomposition des λ -termes en les composantes invariantes que sont les squelettes, enfin, a eu une application plus surprenante avec la preuve de la non-calculabilité des stratégies sans partage optimales pour le λ -calcul faible (théorème 10.6).

Pour conclure sur les méthodes employées dans cette thèse, résumons comment ces dernières nous ont fait voyager d'une région à l'autre dans le monde des CRS. La thèse à proprement parler concerne les CRS faibles orthogonaux, c'est-à-dire la région grisée et numérotée ⑤ dans le diagramme figurant ci-dessous l'ensemble des CRS. Ou plus précisément, elle s'intéresse à la réduction pleinement paresseuse pour ces systèmes, c'est-à-dire aux graphes qui se trouvent au-dessus de cette région ⑤.



Cependant, le principal du travail a été fait ailleurs. D'abord, nous nous sommes appuyés sur des résultats et des constructions de la littérature concernant le λ -calcul (en réduction faible ou non), c'est-à-dire un petit fragment des *CRS* orthogonaux appartenant aux régions ② et ⑤, et concernant les systèmes de réécriture du premier ordre orthogonaux, c'est-à-dire la région ⑦. Ensuite, un certain nombre de propriétés ont été obtenues par instanciation de formalismes abstraits vivant à l'extérieur du monde des *CRS*. Ceci a concerné de nombreuses fois les preuves que les systèmes étiquetés manipulés spécifiaient bien des réductions de graphes, par instanciation du formalisme des systèmes de partage étiquetés *SLS* introduit au chapitre 1. Enfin et surtout, notre progression a été basée sur des transformations et des extensions des systèmes. Et si ces opérations ont toujours été « internalisées » dans le formalisme des *CRS*, elles ne nous en ont pas moins fait voyager d'une région à l'autre.

Pour commencer, la définition des *CRS* étiquetés sous la forme de *CRS* passe par l'opération de raffinement canonique, qui a un impact sur l'éventuelle orthogonalité des systèmes : en partant d'un système orthogonal nous obtenons un système qui n'est qu'« orthogonal en réduction faible » (région ④). La différence n'a pas d'impact sur le comportement du système en réduction faible, mais nous sortons du cadre usuel de l'orthogonalité. En réduction faible nous passons donc de la région ⑤ à la région ④, et effectivement la définition de l'étiquetage pleinement paresseux (chapitre 4) et de l'étiquetage causal (chapitre 7) des *CRS* faibles ainsi que leur comparaison ont eu lieu dans les régions ③ et ④. Seule l'analyse des créations de pas de réduction (lemme 7.7) se passe du raffinement canonique et garde donc un pied dans la région ⑤ (sans y être limitée puisqu'elle ne fait aucune hypothèse d'orthogonalité). En réduction non-faible, l'étiquetage d'un système orthogonal nous fait passer de la région ② à la région ①, comme au chapitre 11. Dans les systèmes du premier ordre enfin le raffinement canonique n'a aucun effet, et nous pouvons donc étiqueter un système orthogonal (région ⑦) sans franchir la frontière de l'orthogonalité.

Ensuite, la traduction au premier ordre par *lifting* permet, pour tout système source en réduction faible, de définir un système cible au premier ordre. La traduction a pour particularité de demander le raffinement canonique préalable du système source. Partant d'un système non-orthogonal de la région ③ cela n'a pas d'impact et nous mène directement dans la région ⑥, à un système du premier ordre non-orthogonal. La situation est en revanche plus intéressante en partant d'un système orthogonal (région ⑤) : le raffinement canonique nous fait d'abord passer une première fois la frontière de l'orthogonalité, nous menant à la région ④ de systèmes orthogonaux en réduction faible. Puis le *lifting* proprement dit nous amène au premier ordre tout en préservant les propriétés d'orthogonalité/de non-orthogonalité du système. Nous obtenons donc un système du premier ordre orthogonal en réduction faible, qui n'est autre qu'un système orthogonal ordinaire : nous passons une seconde fois la frontière de l'orthogonalité pour arriver dans la région ⑦, ce qui permet au chapitre 10 d'invoquer des résultats de la littérature concernant les systèmes du premier ordre orthogonaux.

Enfin, considérons l'ensemble du système dans lequel est étudié le *lifting* généralisé aux *CRS*. Ce système regroupant les réductions source, intermédiaire, et cible ainsi que le *méta-lifting* est toujours un *CRS*. Bien que les réductions source, intermédiaire, et cible soient soumises à la réduction faible, le *méta-lifting* ne l'est pas. De plus, bien que la transformation globale de *lifting* préserve l'éventuelle orthogonalité du système source, le système global n'est en aucun cas orthogonal : le *méta-lifting* forme des recouvrements avec lui-même comme avec la réduction source. Le système global est ainsi non-faible et non-orthogonal, et c'est donc bien à l'intérieur des *CRS* mais depuis la région ①, que nous avons étudié notre transformation allant de la région ⑤ à la région ⑦ en passant par la région ④.

Sélection de questions pour aller plus loin

Dans cette thèse, nous avons cherché à décrire des méthodes d'évaluation à la fois simples et efficaces pour les programmes fonctionnels. Pour cela nous avons d'abord fixé deux contraintes :

- sur le partage, restriction au partage de sous-termes, et
- sur les espaces de réduction, restriction à la réduction faible,

puis nous avons étudié les réductions optimales relativement à ces contraintes, pour notamment conclure au chapitre 10 que le partage de sous-termes était suffisant pour réaliser « l'optimalité faible », c'est-à-dire l'optimalité relativement à l'espace de réduction faible. Cependant, nous avons également constaté au chapitre 11 qu'il était possible d'aller au-delà de l'optimalité faible tout en continuant à respecter la contrainte de ne partager que des sous-termes. Ceci ouvre un premier grand axe de recherche :

*Caractériser les réductions optimales pouvant être réalisées
avec du partage de sous-termes uniquement.*

Cet axe se ramifie en un certain nombre de questions ouvertes concrètes :

- Le partage de sous-termes permet-il d'obtenir des séquences de réduction strictement plus courtes que les réductions optimales sans partage ? Nous avons vu au chapitre 10 que ce n'était pas le cas en réduction faible. En revanche Lamping a montré un exemple en réduction non-faible dans lequel le partage de contextes (en l'occurrence le partage optimal du λ -calcul) permet des séquences de réduction strictement plus courtes que les réductions optimales sans partage [Lam90]. L'association du partage de sous-termes avec la réduction non-faible se situe entre ces deux remarques.
- Le partage de sous-termes suffit-il à exprimer la « paresse complète » de Holst et Gomard [HG91] ? La sémantique naturelle de Sinot pour l'évaluation complètement paresseuse [Sin08] montre que le partage de contextes est utile à sa compréhension. Cependant rien n'indique que le partage de contextes soit nécessaire à la mise en œuvre de cette paresse complète, que Holst et Gomard ont caractérisée en termes d'évaluation partielle et de pleine paresse.

- Le partage de sous-termes permet-il de construire une stratégie de réduction calculable pour le λ -calcul qui génère des séquences de réduction de même longueur que les séquences optimales sans partage ? De même qu'en λ -calcul faible (chapitre 10), en λ -calcul non-faible les stratégies optimales sans partage ne sont pas calculables [BBKV76]. Nous avons vu au chapitre 10 qu'en réduction faible la réduction partagée permettait d'obtenir des stratégies calculables « aussi bonnes ». En réduction non-faible, cela reste à déterminer.

Ces questions théoriques sur l'expressivité de l'évaluation avec partage de sous-termes se doublent de considérations plus pratiques sur la mise en œuvre de tels mécanismes d'évaluation dans le cadre de langages de programmation. Rappelons d'abord que les *CRS*, même faibles et orthogonaux, fournissent un cadre naturellement extensible capable d'exprimer nativement certaines des constructions qui sont usuellement ajoutées de manière *ad hoc* au λ -calcul pour l'étude de langages réels (par exemple OCaml [Ler90] et Haskell [PJS89]). Ainsi, des techniques de mise en œuvre de stratégies de réduction partagée dans certaines sous-classes bien choisies des *CRS* pourraient fournir des mécanismes clé en main pour la mise en œuvre efficace de langages de programmation fonctionnels. Ceci suggère un deuxième axe de recherche future :

*Développer des moyens systématiques pour la mise en œuvre
de stratégies d'évaluation avec partage de sous-termes.*

Nous pouvons développer cet axe en suivant plusieurs questions :

- Déterminer des sous-classes des *CRS* dans lesquelles, partant des règles d'un système et d'une stratégie d'évaluation avec partage, nous pouvons dériver de manière systématique des machines abstraites réalisant les réductions voulues. Nous pouvons ici tirer parti des techniques de *refocusing* de Danvy et Nielsen [DN04] ainsi que de la correspondance fonctionnelle de Danvy et al. entre évaluateurs et machines abstraites [ABDM03]. Cette dernière a d'ailleurs l'avantage d'avoir déjà été appliquée avec succès à des systèmes en appel par nécessité avec partage de sous-termes [PB10, DMMZ10]. Pour appliquer, dans un cadre de *CRS*, de telles techniques basées sur des transformations successives, nous pouvons de plus utiliser des formalisations comme celle développée au chapitre 9 pour généraliser le λ -*lifting*.
- Caractériser des sous-classes de *CRS* et des stratégies d'évaluation pour lesquelles nous pouvons maintenir un lien étroit entre le nombre d'étapes de réduction et le nombre de transitions sur la machine abstraite implémentant le système. Un premier pas a été fait dans cette direction au chapitre 4 avec les résultats de complexité concernant les *CRS* « plats ». Cependant ce premier résultat ne donne qu'un rapport polynomial avec une machine de Turing, c'est-à-dire une machine assez éloignée de la mise en œuvre réelle. Nous pouvons espérer obtenir un rapport de complexité plus étroit avec des machines abstraites plus réalistes et ainsi obtenir,

même au niveau théorique, de meilleurs arguments pour supporter la pertinence de nos optimisations des stratégies de réduction.

Enfin, à la partie III nous avons évoqué un problème théorique fascinant, que nous avons finalement révoqué en nous restreignant à des systèmes orthogonaux : il s'agit de l'étude des dépendances causales entre événements de réduction dans des systèmes non-stables. Cette question des dépendances causales, qui est un support des théories des réductions optimales à la Lévy, porte bien au-delà de l'informatique. Nous pouvons d'ailleurs illustrer la non-stabilité par l'exemple suivant introduit par Wright [Wri88] dans le domaine du droit.

Considérons une maison, et deux incendies X et Y tous deux en passe de détruire cette maison. Si X l'atteint en premier et la brûle, rendant ainsi Y superflu, la responsabilité légale de la destruction de la maison sera sans doute attribuée à X . Pourtant, par la présence de l'incendie Y il est certain que la maison aurait été détruite, même en l'absence de X .

Il y a donc une différence entre être une condition nécessaire ou suffisante au déclenchement d'un événement et en être responsable dans les faits, et l'établissement des causes C d'un événement E dans un tel contexte a posé problème à de nombreux domaines scientifiques. Une tentative de réponse venant de la philosophie est la condition *INUS* de Mackie [Mac65], qui a été simplifiée par Wright [Wri88] en une condition *NESS* (*necessary element of sufficient set*) exprimant que C peut être considéré comme une cause de E si C est un élément nécessaire à la réalisation d'un événement E' qui est lui-même suffisant pour le déclenchement de E . Pearl enfin a proposé une condition de « maintien » combinant ces éléments de suffisance et de nécessité dans une structure plus riche [Pea09] apte à résoudre les paradoxes apportés par les critères de Mackie et Wright.

Ce point apparaît quand un même pas de réduction peut être créé par deux « causes suffisantes indépendantes », comme cela a été vu au chapitre 6 avec la création de deux erreurs indépendantes dans une même opération de filtrage. Ce comportement a pour première conséquence de rendre trop grossière la notion de nécessité qui est à la base de la caractérisation des familles de radicaux, mais vient questionner également les notions de permutation des pas de réduction et de standardisation des séquences de réduction : dans quelle mesure les permutations doivent-elles respecter l'ordre dans lequel adviennent plusieurs causes suffisantes indépendantes de la création d'un même radical ? Un meilleur critère de nécessité peut-il permettre de rendre cette question caduque ?

Explorer la permutation, la standardisation, et les familles de radicaux dans le cas où la dépendance causale n'est plus seulement la nécessité.

Ce dernier axe de recherche, très ouvert, pourrait nous mener à une compréhension bien plus profonde de ce qu'est le calcul. Au-delà du calcul séquentiel que nous avons considéré dans cette thèse, il touche aussi le calcul concurrent dans lequel les causes indépendantes d'un même événement peuvent survenir très naturellement. De plus, cet axe est connecté à de nombreux autres domaines scientifiques, qui sont autant d'aides potentielles à son développement.

Bibliographie

- [ABDM03] Mads Sig Ager, Dariusz Biernacki, Olivier Danvy, and Jan Midtgaard. A functional correspondence between evaluators and abstract machines. In *PPDP*, pages 8–19, 2003. <http://doi.acm.org/10.1145/888251.888254>.
- [ACCL90] Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. In *POPL*, pages 31–46, 1990. <http://doi.acm.org/10.1145/96709.96712>.
- [Acz78] Peter Aczel. A general church-rosser theorem. Technical report, University of Manchester, 1978.
- [AF97] Zena M. Ariola and Matthias Felleisen. The call-by-need lambda calculus. *Journal of Functional Programming*, 7(3) :265–301, 1997. <http://journals.cambridge.org/action/displayAbstract?aid=44089>.
- [AFM⁺95] Zena M. Ariola, Matthias Felleisen, John Maraist, Martin Odersky, and Philip Wadler. The call-by-need lambda calculus. In *POPL*, pages 233–246, 1995. <http://doi.acm.org/10.1145/199448.199507>.
- [AG98] Andrea Asperti and Stefano Guerrini. *The Optimal Implementation of Functional Programming Languages*, volume 45. Cambridge University Press, 1998.
- [AK96] Zena M. Ariola and Jan Willem Klop. Equational term graph rewriting. *Fundamenta Informaticae*, 26(3/4) :207–240, 1996. <http://dx.doi.org/10.3233/FI-1996-263401>.
- [AL96] Andrea Asperti and Cosimo Laneve. Interaction systems ii : The practice of optimal reductions. *Theoretical Computer Science*, 159(2) :191–244, 1996. [http://dx.doi.org/10.1016/0304-3975\(95\)00062-3](http://dx.doi.org/10.1016/0304-3975(95)00062-3).
- [AM01] Andrea Asperti and Harry G. Mairson. Parallel beta reduction is not elementary recursive. *Information and Computation*, 170(1) :49–80, 2001. <http://dx.doi.org/10.1006/inco.2001.2869>.
- [AM10] Martin Avanzini and Georg Moser. Closing the gap between runtime complexity and polytime computability. In *RTA*, pages 33–48, 2010. <http://dx.doi.org/10.4230/LIPIcs.RTA.2010.33>.

- [Amt93] Torben Amtoft. *Sharing of Computations*. Ph.D thesis, University of Aarhus, 1993.
- [Bal10a] Thibaut Balabonski. Optimality for dynamic patterns. In *PPDP*, pages 231–242, 2010. <http://doi.acm.org/10.1145/1836089.1836119>.
- [Bal10b] Thibaut Balabonski. Optimality for Dynamic Patterns. <http://hal.archives-ouvertes.fr/hal-00508393/PDF/Balabonski-OfDP.pdf>, March 2010.
- [Bal12a] Thibaut Balabonski. Axiomatic sharing-via-labelling. In *RTA*, pages 85–100, 2012. <http://dx.doi.org/10.4230/LIPIcs.RTA.2012.85>.
- [Bal12b] Thibaut Balabonski. A unified approach to fully lazy sharing. In *POPL*, pages 469–480, 2012. <http://doi.acm.org/10.1145/2103656.2103713>.
- [Bar84] Hendrik Pieter Barendregt. *The Lambda Calculus – Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company, Amsterdam, revised edition edition, 1984.
- [BBCK07] Paolo Baldan, Clara Bertolissi, Horatiu Cirstea, and Claude Kirchner. A rewriting calculus for cyclic higher-order term graphs. *Mathematical Structures in Computer Science*, 17(3) :363–406, 2007. <http://dx.doi.org/10.1017/S0960129507006093>.
- [BBKV76] Hendrik Pieter Barendregt, Jan A. Bergstra, Jan Willem Klop, and Henri Volken. Some notes on lambda reduction. Technical Report 22, University of Utrecht, Department of mathematics, 1976.
- [Ber78] Gérard Berry. Stable models of typed lambda-calculi. In *ICALP*, pages 72–89, 1978. http://dx.doi.org/10.1007/3-540-08860-1_7.
- [BKKS87] Hendrik Pieter Barendregt, Richard Kennaway, Jan Willem Klop, and M. Ronan Sleep. Needed reduction and spine strategies for the lambda calculus. *Information and Computation*, 75(3) :191–231, 1987. [http://dx.doi.org/10.1016/0890-5401\(87\)90001-0](http://dx.doi.org/10.1016/0890-5401(87)90001-0).
- [BKLR12] Eduardo Bonelli, Delia Kesner, Carlos Lombardi, and Alejandro Ríos. Normalisation for dynamic pattern calculi. In *RTA*, pages 117–132, 2012. <http://dx.doi.org/10.4230/LIPIcs.RTA.2012.117>.
- [BLM05] Tomasz Blanc, Jean-Jacques Lévy, and Luc Maranget. Sharing in the weak lambda-calculus. In *Processes, Terms and Cycles*, pages 70–87, 2005. http://dx.doi.org/10.1007/11601548_7.
- [BLM07] Tomasz Blanc, Jean-Jacques Lévy, and Luc Maranget. Sharing in the weak lambda-calculus revisited. In Erik Barendsen, Herman Geuvers, Venanzio Capretta, and Milad Niqui, editors, *Reflections on Type Theory, Lambda Calculus, and the Mind*, pages 41–50. ICIS,

- Faculty of Science, Radboud University Nijmegen, 2007. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday.
- [Blo01] Stefan Blom. *Term Graph Rewriting - Syntax and Semantics*. Ph.D. thesis, Vrije Universiteit, Amsterdam, 2001.
- [BvEG⁺87] Hendrik Pieter Barendregt, Marko C. J. D. van Eekelen, John R. W. Glauert, Richard Kennaway, Marinus J. Plasmeijer, and M. Ronan Sleep. Term graph rewriting. In *PARLE (2)*, pages 141–158, 1987. http://dx.doi.org/10.1007/3-540-17945-3_8.
- [CF74] Haskell B. Curry and Robert Feys. *Combinatory Logic*, volume I. North-Holland Publishing Company, Amsterdam, third printing edition, 1974.
- [CF12] Stephen Chang and Matthias Felleisen. The call-by-need lambda calculus, revisited. In *ESOP*, pages 128–147, 2012. http://dx.doi.org/10.1007/978-3-642-28869-2_7.
- [cH98] Naim Çagman and J. Roger Hindley. Combinatory weak reduction in lambda calculus. *Theoretical Computer Science*, 198(1-2) :239–247, 1998. [http://dx.doi.org/10.1016/S0304-3975\(97\)00250-8](http://dx.doi.org/10.1016/S0304-3975(97)00250-8).
- [Chu32] Alonzo Church. A set of postulates for the foundation of logic. *The Annals of Mathematics*, 33(2) :346–366, 1932. <http://www.jstor.org/stable/1968337>.
- [DLLL05] Daniel J. Dougherty, Pierre Lescanne, Luigi Liquori, and Frédéric Lang. Addressed term rewriting systems : Syntax, semantics, and pragmatics : Extended abstract. *Electronic Notes in Theoretical Computer Science*, 127(5) :57–82, 2005. <http://dx.doi.org/10.1016/j.entcs.2004.12.042>.
- [DMMZ10] Olivier Danvy, Kevin Millikin, Johan Munk, and Ian Zerny. Defunctionalized interpreters for call-by-need evaluation. In *FLOPS*, pages 240–256, 2010. http://dx.doi.org/10.1007/978-3-642-12251-4_18.
- [DN01] Olivier Danvy and Lasse R. Nielsen. Defunctionalization at work. In *PPDP*, pages 162–174, 2001. <http://doi.acm.org/10.1145/773184.773202>.
- [DN04] Olivier Danvy and Lasse R. Nielsen. Refocusing in reduction semantics, 2004.
- [DS00] Olivier Danvy and Ulrik Pagh Schultz. Lambda-dropping : transforming recursive equations into programs with block structure. *Theoretical Computer Science*, 248(1-2) :243–287, 2000. [http://dx.doi.org/10.1016/S0304-3975\(00\)00054-2](http://dx.doi.org/10.1016/S0304-3975(00)00054-2).
- [dSE43] Antoine de Saint-Exupéry. *Le Petit Prince*. Gallimard, 1943.
- [FM98] Maribel Fernández and Ian Mackie. Interaction nets and term-rewriting systems. *Theoretical Computer Science*, 190(1) :3–39, 1998. [http://dx.doi.org/10.1016/S0304-3975\(97\)00082-0](http://dx.doi.org/10.1016/S0304-3975(97)00082-0).

- [FM01] Fabrice Le Fessant and Luc Maranget. Optimizing pattern matching. In *ICFP*, pages 26–37, 2001. <http://doi.acm.org/10.1145/507635.507641>.
- [FMS05] Maribel Fernández, Ian Mackie, and François-Régis Sinot. Closed reduction : explicit substitutions without alpha-conversion. *Mathematical Structures in Computer Science*, 15(2) :343–381, 2005. <http://dx.doi.org/10.1017/S0960129504004633>.
- [FMSW09] Maribel Fernández, Ian Mackie, Shinya Sato, and Matthew Walker. Recursive functions with pattern matching in interaction nets. *Electronic Notes in Theoretical Computer Science*, 253(4) :55–71, 2009. <http://dx.doi.org/10.1016/j.entcs.2009.10.017>.
- [FW76] Daniel P. Friedman and David S. Wise. Cons should not evaluate its arguments. In *ICALP*, pages 257–284, 1976.
- [GAL92] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In *POPL*, pages 15–26, 1992. <http://doi.acm.org/10.1145/143165.143172>.
- [GK96] John R. W. Glauert and Zurab Khasidashvili. Relative normalization in deterministic residual structures. In *CAAP*, pages 180–195, 1996. http://dx.doi.org/10.1007/3-540-61064-2_37.
- [Gue96] Stefano Guerrini. *Theoretical and Practical Issues of Optimal Implementations of Functional Languages*. Ph.D. thesis, Università di Pisa, 1996.
- [Has97] Masahito Hasegawa. *Models of Sharing Graphs*. Ph.D. thesis, University of Edinburgh, 1997.
- [HG91] Carsten Kehler Holst and Carsten K. Gomard. Partial evaluation is fuller laziness. In *PEPM*, pages 223–233, 1991. <http://doi.acm.org/10.1145/115865.115890>.
- [Hin77] J. Roger Hindley. Combinatory reductions and lambda reductions compared. *Mathematical Logic Quarterly*, 23(7-12) :169–180, 1977. <http://dx.doi.org/10.1002/malq.19770230708>.
- [HJ76] Peter Henderson and James H. Morris Jr. A lazy evaluator. In *POPL*, pages 95–103, 1976. <http://doi.acm.org/10.1145/800168.811543>.
- [HJ03] Ralf Hinze and Johan Jeuring. Generic haskell : Practice and theory. In *Generic Programming*, pages 1–56, 2003. http://dx.doi.org/10.1007/978-3-540-45191-4_1.
- [How70] William Alvin Howard. Assignment of ordinals to terms for primitive recursive functionals of finite type. In *Intuitionism and Proof Theory*, pages 443–458, 1970.
- [Hug82] R. J. M. Hughes. Super combinators : A new implementation method for applicative languages. In *LFP*, pages 1–10, 1982.

- [Jay04] C. Barry Jay. The pattern calculus. *ACM Transactions on Programming Languages and Systems*, 26(6) :911–937, 2004. <http://doi.acm.org/10.1145/1034774.1034775>.
- [Jay09] C. Barry Jay. *Pattern Calculus : Computing with Functions and Data Structures*. Springer, 2009.
- [JK06] C. Barry Jay and Delia Kesner. Pure pattern calculus. In *ESOP*, pages 100–114, 2006. http://dx.doi.org/10.1007/11693024_8.
- [JK08] C. Barry Jay and Delia Kesner. Patterns as First-Class Citizens. Technical report, 2008. Available as <http://hal.archives-ouvertes.fr/hal-00229331/fr/>.
- [JK09] C. Barry Jay and Delia Kesner. First-class patterns. *Journal of Functional Programming*, 19(2) :191–225, 2009. <http://dx.doi.org/10.1017/S0956796808007144>.
- [Joh85] Thomas Johnsson. Lambda lifting : Treansforming programs to recursive equations. In *FPCA*, pages 190–203, 1985. http://dx.doi.org/10.1007/3-540-15975-4_37.
- [Kah98] Wolfram Kahl. Relational treatment of term graphs with bound variables. *Logic Journal of the IGPL*, 6(2) :259–303, 1998. <http://dx.doi.org/10.1093/jigpal/6.2.259>.
- [Kah99] Wolfram Kahl. The term graph programming system hops. In *Tool Support for System Specification, Development and Verification, Advances in Computing Science*, pages 136–149, 1999.
- [Kat90] Vinod Kathail. *Optimal interpreters for lambda-calculus based functional languages*. Ph.D. thesis, Massachusetts Institute of Technology, 1990. <http://hdl.handle.net/1721.1/14040>.
- [Kes07] Delia Kesner. The theory of calculi with explicit substitutions revisited. In *CSL*, pages 238–252, 2007. http://dx.doi.org/10.1007/978-3-540-74915-8_20.
- [Kha93] Zurab Khasidashvili. Optimal normalization in orthogonal term rewriting systems. In *RTA*, pages 243–258, 1993. http://dx.doi.org/10.1007/3-540-56868-9_19.
- [Klo80] Jan Willem Klop. *Combinatory Reduction Systems*. Ph.D. thesis, Mathematical Centre Tracts Nr. 127. CWI, Amsterdam, 1980.
- [Kri07] Jean-Louis Krivine. A call-by-name lambda-calculus machine. *Higher-Order and Symbolic Computation*, 20(3) :199–207, September 2007. <http://dx.doi.org/10.1007/s10990-007-9018-9>.
- [KvOdV08] Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Lambda calculus with patterns. *Theoretical Computer Science*, 398(1-3) :16–31, 2008. <http://dx.doi.org/10.1016/j.tcs.2008.01.019>.

- [KvOvR93] Jan Willem Klop, Vincent van Oostrom, and Femke van Raamsdonk. Combinatory reduction systems : Introduction and survey. *Theoretical Computer Science*, 121(1&2) :279–308, 1993. [http://dx.doi.org/10.1016/0304-3975\(93\)90091-7](http://dx.doi.org/10.1016/0304-3975(93)90091-7).
- [Lam90] John Lamping. An algorithm for optimal lambda calculus reduction. In *POPL*, pages 16–30, 1990. <http://doi.acm.org/10.1145/96709.96711>.
- [Lau93] John Launchbury. A natural semantics for lazy evaluation. In *POPL*, pages 144–154, 1993. <http://doi.acm.org/10.1145/158511.158618>.
- [Ler90] Xavier Leroy. The ZINC experiment : an economical implementation of the ML language. Technical report 117, INRIA, 1990.
- [LM06] Ugo Dal Lago and Simone Martini. An invariant cost model for the lambda calculus. In *CiE*, pages 105–114, 2006. http://dx.doi.org/10.1007/11780342_11.
- [LM09a] Ugo Dal Lago and Simone Martini. Derivational complexity is an invariant cost model. In *FOPARA*, pages 100–113, 2009. http://dx.doi.org/10.1007/978-3-642-15331-0_7.
- [LM09b] Ugo Dal Lago and Simone Martini. On constructor rewrite systems and the lambda-calculus. In *ICALP (2)*, pages 163–174, 2009. http://dx.doi.org/10.1007/978-3-642-02930-1_14.
- [Lé78] Jean-Jacques Lévy. *Réductions correctes et optimales dans le lambda-calcul*. Ph.D. thesis, Université Paris VII, 1978.
- [Lé80] Jean-Jacques Lévy. Optimal reductions in the lambda-calculus. In *To H.B. Curry : Essays on Combinatory Logic, Lambda Calculus and Formalisms*, pages 159–191, 1980.
- [Mac65] J. L. Mackie. Causes and conditions. *American Philosophical Quarterly*, 2(4) :245–264, 1965. <http://www.jstor.org/stable/20009173>.
- [Mac98] Ian Mackie. Yale : Yet another lambda evaluator based on interaction nets. In *ICFP*, pages 117–128, 1998. <http://doi.acm.org/10.1145/289423.289434>.
- [Mac04] Ian Mackie. Efficient lambda-evaluation with interaction nets. In *RTA*, pages 155–169, 2004. http://dx.doi.org/10.1007/978-3-540-25979-4_11.
- [Mar91] Luc Maranget. Optimal derivations in weak lambda-calculi and in orthogonal terms rewriting systems. In *POPL*, pages 255–269, 1991. <http://doi.acm.org/10.1145/99583.99618>.
- [Mar92] Luc Maranget. *La stratégie paresseuse*. Ph.D. thesis, Université Paris VII, 1992.
- [Mar08] Luc Maranget. Compiling pattern matching to good decision trees. In *ML*, pages 35–46, 2008. <http://doi.acm.org/10.1145/1411304.1411311>.

- [Mel96] Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture*. Ph.D. thesis, 1996.
- [Mel05] Paul-André Melliès. Axiomatic rewriting theory i : A diagrammatic standardization theorem. In *Processes, Terms and Cycles*, pages 554–638, 2005. http://dx.doi.org/10.1007/11601548_23.
- [MOW98] John Maraist, Martin Odersky, and Philip Wadler. The call-by-need lambda calculus. *Journal of Functional Programming*, 8(3) :275–317, 1998. <http://journals.cambridge.org/action/displayAbstract?aid=44169>.
- [Nip91] Tobias Nipkow. Higher-order critical pairs. In *LICS*, pages 342–349, 1991. <http://dx.doi.org/10.1109/LICS.1991.151658>.
- [PB10] Maciej Piróg and Dariusz Biernacki. A systematic derivation of the stg machine verified in coq. In *Haskell*, pages 25–36, 2010. <http://doi.acm.org/10.1145/1863523.1863528>.
- [Pea09] Judea Pearl. *Causality*. Cambridge University Press, second edition edition, 2009.
- [PJ87] Simon L. Peyton-Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall, 1987.
- [PJS89] Simon L. Peyton-Jones and Jon Salkild. The spineless tagless g-machine. In *FPCA*, pages 184–201, 1989. <http://doi.acm.org/10.1145/99370.99385>.
- [Plu99] Detleif Plump. Term graph rewriting. 2 :3–61, 1999.
- [Rey98] John C. Reynolds. Definitional interpreters for higher-order programming languages. *Higher-Order and Symbolic Computation*, 11(4) :363–397, 1998. <http://dx.doi.org/10.1023/A:1010027404223>.
- [Ros96] Kristoffer Høgsbro Rose. Explicit substitution – tutorial & survey. Lecture Series LS–96–3, BRICS, 1996. <ftp://ftp.brics.dk/LS/96/3/>.
- [Sco68] Dana Scott. A system of functional abstraction, 1968. Lectures delivered at University of California, Berkeley, Cal., 1962/63.
- [Ses97] Peter Sestoft. Deriving a lazy abstract machine. *Journal of Functional Programming*, 7(3) :231–264, 1997. <http://journals.cambridge.org/action/displayAbstract?aid=44087>.
- [Sin08] François-Régis Sinot. Complete laziness : a natural semantics. *Electronic Notes in Theoretical Computer Science*, 204 :129–145, 2008. <http://dx.doi.org/10.1016/j.entcs.2008.03.058>.
- [SR90] R. C. Sekar and I. V. Ramakrishnan. Programming in equational logic : Beyond strong sequentiality. In *LICS*, pages 230–241, 1990. <http://dx.doi.org/10.1109/LICS.1990.113749>.
- [SW10] Olin Shivers and Mitchell Wand. Bottom-up beta-reduction : Uplinks and lambda-dags. *Fundamenta Informaticae*, 103(1-4) :247–287, 2010. <http://dx.doi.org/10.3233/FI-2010-328>.

- [Ter03] Terese. *Term Rewriting Seminar – Terese*. Vol. 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.
- [Thy99] Michael Jonathan Thyer. *Lazy Specialization*. Ph.D. thesis, York, 1999.
- [Tur79] David A. Turner. A new implementation technique for applicative languages. *Software : Practice and Experience*, 9(1) :31–49, 1979.
- [vO97] Vincent van Oostrom. Finite family developments. In *RTA*, pages 308–322, 1997. http://dx.doi.org/10.1007/3-540-62950-5_80.
- [vOvR11] Vincent van Oostrom and Femke van Raamsdonk, 2011. Communication privée.
- [Wad71] Christopher P. Wadsworth. *Semantics and Pragmatics of the Lambda Calculus*. Ph.D. thesis, Oxford, 1971.
- [Wri88] Richar W. Wright. Causation, responsibility, risk, probability, naked statistics, and proof : Pruning the bramble bush by clarifying the concepts. *Iowa Law Review*, (73) :1001–1077, 1988.
- [Yos94] Nobuko Yoshida. Optimal reduction in weak- λ -calculus with shared environments. *Journal of Computer Software*, 11(5) :2–20, 1994.

Table des matières

Introduction	7
Matériel	35
0.1 Relations binaires	35
0.2 Réécriture abstraite	36
0.3 Signatures, termes, contextes	41
0.4 Réécriture du premier ordre	42
0.5 Lambda-calcul	44
0.6 <i>Combinatory Reduction Systems</i>	48
I Partage	55
1 Un cadre axiomatique pour le partage	57
1.1 Systèmes abstraits de réécriture de termes	59
1.2 Des étiquettes pour un exemple informel	62
1.3 Le cadre axiomatique	65
1.3.1 <i>ATRS Marqués</i>	65
1.3.2 <i>Systèmes de partage étiquetés (SLS)</i>	66
1.4 Exemple formel	69
1.5 Comparaison d'étiquetages	71
1.6 Discussion	73
1.6.1 Expressivité du cadre axiomatique	73
1.6.2 Comparaison avec d'autres techniques	74
1.6.3 Utilisabilité du cadre axiomatique	76
II Pleine paresse	81
2 L'algorithme de Wadsworth	83
2.1 Les λ -graphes sur un exemple	84
2.2 Structure du λ -calcul faible	86
2.3 Représentation étiquetée des λ -graphes	90
2.4 Discussion : appel par nécessité	94
2.4.1 Stratégies de réduction	94
2.4.2 Représentation des variables liées	94

3 Les héritiers de Wadsworth	97
3.1 Tour d'horizon de la littérature	98
3.1.1 Techniques générales pour le partage de sous-termes	98
3.1.2 Tour d'horizon de la pleine paresse	99
3.2 Un étiquetage paramétrique pour le λ -calcul	101
3.2.1 <i>SLS</i> pour les β -systèmes parcimonieux	102
3.2.2 <i>SLS</i> pour les β -systèmes généreux	104
3.3 Retour à la littérature	107
3.4 La famille pleinement paresseuse	110
3.5 Discussion : partage	112
3.5.1 Une combinaison pathologique	112
3.5.2 Renommage et partage	114
4 Généralisation à l'ordre supérieur	119
4.1 Réduction faible dans les <i>CRS</i>	121
4.2 Raffinements	124
4.3 Définition du partage pleinement paresseux	133
4.4 Comparaison avec le λ -calcul étiqueté	136
4.5 Discussion : complexité	139
4.5.1 Une mesure de croissance	140
4.5.2 Complexité dérivationnelle	141
III Causalité	147
5 Le cas du λ-calcul	149
5.1 Causalité dans le λ -calcul	150
5.2 Causalité en réduction faible	153
5.3 Lambda-calcul faible étiqueté	155
5.3.1 Définition	155
5.3.2 Propriété de partage	158
5.4 Correction et complétude causales	160
5.5 Discussion : variantes de forme	163
5.5.1 Contribution et ordre <i>SLS</i>	164
5.5.2 Les positions des étiquettes	164
5.5.3 Lien avec la pleine paresse	165
5.6 Discussion : causalités	166
5.6.1 Des causes du dégel	166
5.6.2 Causalité retardée	168
5.6.3 Causalité anticipée	170
5.7 Discussion : reconstruction	170
5.7.1 Reconstruction ambiguë	171
5.7.2 Retrouver la propriété de reconstruction	172
5.8 Discussion : une autre preuve de partage	174
5.8.1 Une preuve causale du partage	174
5.8.2 Éloignement de la causalité	175

6	Histoire d'un calcul non-séquentiel	177
6.1	Filtrage et motifs dynamiques	178
6.2	<i>Pure Pattern Calculus</i>	181
6.3	Causalité dans le <i>Pure Pattern Calculus</i>	187
6.4	Filtrage étiqueté	192
6.5	<i>Pure Pattern Calculus</i> faible étiqueté	199
6.5.1	Définition	199
6.5.2	Propriété de partage	201
6.6	Propriétés causales	203
6.7	Discussion : non-stabilité	215
6.7.1	Instabilité des noms	215
6.7.2	Non-stabilité et nécessité	217
6.7.3	Stabilisation	218
7	Étiquetage causal des CRS faibles	221
7.1	Causalité dans les <i>CRS</i>	221
7.1.1	Retour sur la création et la résiduation	222
7.1.2	Création et modification de radicaux	224
7.2	Étiquetage	228
7.2.1	Expansion	230
7.2.2	Étiquetage	231
7.3	Équivalence avec le partage pleinement paresseux	234
7.3.1	Propriété de partage	234
7.3.2	Pleine paresse avec expansion	236
7.3.3	De la transparence de l'expansion	237
7.4	Propriétés de contribution	241
7.4.1	Contribution directe	241
7.4.2	Instabilité des radicaux	242
7.5	Correction et complétude causales	244
IV	Premier ordre	249
8	<i>Lambda-lifting</i> pleinement paresseux	251
8.1	Extractions	252
8.2	<i>Lambda-lifting</i>	254
8.3	Bisimulation	264
8.4	Discussion : réduction faible et orthogonalité	268
8.5	<i>Lambda-lifting</i> étiqueté	269
9	<i>Lifting</i> généralisé	279
9.1	<i>Lifting</i> généralisé	280
9.2	Réductions source, intermédiaire, et cible	286
9.3	Bisimulation	291
9.4	Application à un système étiqueté	295
9.4.1	Définition d'un système de <i>lifting</i> étiqueté stable.	296

9.4.2	Préservation des descendance	299
9.4.3	Préservation du partage	301
V	Optimalités	307
10	Optimalité faible	309
10.1	Optimalité faible sans partage	310
10.2	Non-calculabilité des stratégies optimales	312
10.3	Partage et effectivité	319
10.4	Un cadre axiomatique pour les familles	321
10.5	Optimalité faible	325
11	Au-delà de l'optimalité faible	329
11.1	Sous-optimalité des familles faibles	330
11.2	Partage optimisé	333
11.3	Partage en réduction non-faible	337
11.4	Réduction (non-)faible et (non-)orthogonalité	340
11.5	Zones d'effet optimisées	343
11.6	Un dernier exemple	344
	Conclusion générale	349

Cette thèse explore des stratégies d'évaluation efficaces pour les langages de programmation fonctionnelle et cherche à déterminer les meilleures stratégies atteignables dans un cadre de réécriture d'ordre supérieur. Pour une meilleure adéquation avec les mécanismes de base communément mis en œuvre dans les langages fonctionnels, le modèle théorique usuel est soumis à deux contraintes : pas d'évaluation anticipée du corps non-instancié d'une fonction, et une restriction des mécanismes de partage à des sous-termes complets.

Dans le cadre du lambda-calcul ainsi que dans celui plus général de la réécriture du second ordre orthogonale, trois modèles d'évaluation sont démontrés équivalents en réduction faible : la réduction optimale sans partage, la réduction pleinement paresseuse à la Wadsworth, et la réduction optimale de familles à la Lévy. Ceci définit une notion unifiée d'optimalité faible, qui nous apprend que l'optimalité à la Lévy en réduction faible peut être réalisée aussi bien par une stratégie avec partage pleinement paresseuse que par une stratégie sans partage. Il est toutefois aussi démontré qu'une telle stratégie optimale sans partage ne peut être calculable, alors que le partage permet dans de nombreux cas d'obtenir une stratégie optimale très simple.

Pour établir ces résultats, cette thèse tire parti des restrictions imposées pour se ramener à des systèmes plus simples et bien maîtrisés. D'une part les graphes modélisant le partage de sous-termes sont systématiquement représentés par des termes étiquetés, grâce au cadre axiomatique des systèmes de partage étiquetés (Sharing-via-Labeling Systems, SLS) développé pour l'occasion. D'autre part, une transformation inspirée du lambda-lifting établit une bisimulation entre tout système du second ordre faible et un système du premier ordre, bisimulation qui préserve entre autres la longueur des séquences de réduction, l'éventuelle orthogonalité des systèmes, et le partage. Des graphes pour la réécriture d'ordre supérieur sont ainsi formalisés et étudiés, sans graphes ni ordre supérieur.

This dissertation investigates efficient evaluation strategies for functional programming languages and aims at determining the best reachable strategies using a higher-order rewriting framework. The adequacy of the usual theoretical model with respect to the base mechanisms commonly implemented in functional languages is improved by considering two constraints : no early evaluation of non-instantiated function bodies, and a restriction of sharing mechanisms to complete subterms.

In the lambda-calculus, as well as in the more general framework of orthogonal second-order rewriting, three evaluation models are considered : optimal reduction without sharing, fully lazy reduction à la Wadsworth, and optimal family reduction à la Lévy. This defines a unified notion of weak optimality, that teaches us that Lévy-optimality in weak reduction can be implemented by a fully lazy strategy as well as a strategy without sharing. It is however also proved that such an optimal strategy without sharing cannot be computable, while sharing allows in a number of cases the definition of very simple optimal strategies.

To reach these conclusions, this dissertation makes good use of the constraints to reduce the discussion to simpler well-known systems. On the one hand the graphs modelling the sharing of subterms are represented by labelled terms, thanks to the axiomatic framework of Sharing-via-Labeling Systems (SLS) developed for the occasion. On the other hand, a transformation process inspired from lambda-lifting defines a bisimulation between any weak second-order system and some first-order system, said bisimulation preserving in particular the length of reduction sequences, the possible orthogonality of the systems, and sharing. Thus graphs for higher-order rewriting are formalized and studied, without graphs nor higher-order.