



AIX-MARSEILLE UNIVERSITÉ
ED 184 : MATHÉMATIQUES ET INFORMATIQUE
UFR SCIENCES
INSTITUT DE MATHÉMATIQUES DE MARSEILLE

THÈSE

présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ D'AIX-MARSEILLE

Discipline : Mathématiques

par

Jean-Baptiste MIDEZ

Une étude combinatoire du λ -calcul avec ressources uniforme

Soutenue le 15 décembre 2014

devant le jury composé de :

| | | | |
|---------|------------------|--------------------------|--------------------|
| Daniel | HIRSCHKOFF | ENS Lyon | Rapporteur |
| Lorenzo | TORTORA de FALCO | Université Rome 3 | Rapporteur |
| Thomas | EHRHARD | Université Paris 7 | Examineur |
| Michele | PAGANI | Université Paris 7 | Examineur |
| Lionel | VAUX | Université Aix-Marseille | Examineur |
| Laurent | REGNIER | Université Aix-Marseille | Directeur de thèse |

Remerciements

Je voudrais tout d'abord remercier Laurent Regnier pour avoir accepté de diriger cette thèse. Il a su me redonner confiance lorsque je butais sur un problème grâce à son enthousiasme sans faille à chaque nouveau résultat, si petit soit-il. Je le remercie également pour la liberté qu'il m'a laissé dans mes recherches ainsi que pour tous les échanges constructifs que nous avons eu.

Je voudrais également remercier Daniel Hirschhoff et Lorenzo Tortora de Falco pour avoir accepté d'être rapporteurs de cette thèse.

Je souhaite remercier Thomas Ehrhard, Michele Pagani et Lionel Vaux pour avoir accepté de faire partie de mon jury.

Je remercie Laurent Regnier et Daniel Hirschhoff pour leur relecture attentive et pour toutes les remarques qu'ils m'ont faites.

Je remercie toute l'équipe de logique pour leur disponibilité et pour la bonne ambiance qui règne au sein de l'équipe : Myriam Quatrini (pour sa bonne humeur), Lionel Vaux et Emmanuel Beffara (pour leur aide informatique et pour leur grande disponibilité), Yves Lafont et bien sûr Laurent Regnier. Sans oublier tous les doctorants et post-doc qui sont passés par là pendant mon séjour : Michele Alberti, Pierre Rannou, Matteo Acclavio, Eugenia Sironi, Marc Bagnol, Paolo Pistone, Etienne Dushesne.

Je voudrais remercier particulièrement Thomas Seiller pour son accueil lors de mon stage de Master 2, pour l'intérêt qu'il a porté à mes recherches, pour tous les contre-exemples qu'il m'a donné et les nombreuses discussions que nous avons eu que ce soit sur mes recherches ou sur d'autres sujets, pour m'avoir proposé de participer à son projet "Vérité et Preuves" qui a été très intéressant et pour avoir organisé les "Séances de Mathématiques" qui ont été très instructives.

Je souhaite également remercier l'ensemble des doctorants pour tous les bons moments passés en leur compagnie, les petits-déjeuners, les parties de belote,

les pauses... : Marc Bagnol, Eugenia Sironi, Michele Alberti, Paolo Pistone, Pierre-Olivier Goffard, Marc Munsch, Annamaria Iezzi, Florent Rovetta, Julia Pieltant, Hamish Ivey-Law, Virgile Ducet, Marcelo Gonçalves de Martino, Sarah Bray, Émilie Delnieppe, Pierre Lofredi.

Je tiens à remercier particulièrement Joël Cohen pour toutes les discussions que nous avons eu, pour avoir été là quand j'en avais besoin, pour ses conseils et pour tant d'autres choses.

Je remercie Christian Maire et Pierre Arnoux pour m'avoir encadré respectivement lors de mon stage de L3 et de M1 et qui m'ont donné envie de faire de la recherche, ainsi que Laurent Regnier pour m'avoir encadré pour mon stage de M2 sans lequel je n'aurai pas fait cette thèse.

Je voudrais remercier mes professeurs de mathématiques de collège et de lycée pour m'avoir donné le goût des mathématiques.

Je remercie mes amis pour leur soutien : Anne, Jean-Paul, Isabelle, Julien et Joël.

Je souhaite remercier tous les musiciens des diverses formations musicales auxquelles j'ai participé au cours de ma thèse pour m'avoir permis de continuer à jouer de la musique.

Je remercie également mes parents pour m'avoir poussé à continuer mes études et plus particulièrement mon père pour avoir été aussi pinailleur, mon frère Théophile et sa femme Caroline parce qu'ils m'ont toujours soutenu et encouragé, ainsi que toute ma famille.

Je remercie Archi pour m'avoir tenu compagnie tout au long de cette thèse.

Enfin je remercie Claire pour son soutien, sa compréhension et sa patience.

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 7 |
| 2 | Préliminaires | 11 |
| 2.1 | Notations | 11 |
| 2.2 | Multi-ensembles | 11 |
| 2.3 | Le λ -calcul pur | 12 |
| 3 | Le λ-calcul avec ressources | 15 |
| 3.1 | β -réduction | 17 |
| 3.2 | Γ -réduction | 28 |
| 3.3 | β -expansion | 36 |
| 4 | Relations familiales | 41 |
| 4.1 | Frères | 42 |
| 4.2 | Cousins | 43 |
| 4.3 | Les termes sans λ et sans monôme vide | 46 |
| 4.4 | Les termes sans monômes vides | 48 |
| 4.5 | Les termes dont tous les monômes ont la même taille | 49 |
| 4.6 | Décidabilité du cousinage | 50 |
| 4.7 | Musée des horreurs | 50 |
| 5 | Programmation | 53 |
| 5.1 | Forme normale | 54 |
| 5.2 | β -expansion | 55 |
| 5.3 | Cousins | 56 |
| 5.4 | Correspondance entre la β -réduction et la Γ -réduction | 57 |
| 6 | Conclusion | 59 |
| 6.1 | Travaux futurs | 59 |

Chapitre 1

Introduction

Le λ -calcul est un langage formel défini dans les années trente par Alonzo Church afin de caractériser les fonctions récursives. Ce langage extrêmement simple par sa définition, seulement deux constructeurs, est pourtant un outil de calcul très complexe. C'est aussi un précurseur des langages de programmation. On le retrouve aujourd'hui derrière la plupart des langages de programmation fonctionnelle et des outils de démonstration assisté par ordinateur, d'où son importance actuelle.

Dans les années 80, Girard a défini une logique «linéaire» ([Gir87]). Dans la logique classique, démontrer la proposition « A implique B » revient à démontrer B en utilisant la démonstration de A . Mais on ne spécifie pas le nombre de fois dont on pourrait avoir besoin d'utiliser la preuve de A pour démontrer B . La logique linéaire permet de faire cette distinction et elle se dote de constructeurs supplémentaires pour en rendre compte. Notamment le constructeur «!» (bien sûr) qui, apposé devant une formule, signifie que cette formule peut être utilisée autant de fois que nécessaire. Sans ce constructeurs une formule ne peut être utilisée qu'une et une seule fois. Cette logique linéaire permet alors d'étudier la façon dont sont utilisées les formules au cours d'une preuve et notamment combien de fois elles sont utilisées. D'autre part elle sert à typer les programmes d'ordre supérieur, ceux qui peuvent être représenté par le λ -calcul typé. Elle permet alors d'étudier la façon dont ces programmes gèrent leurs ressources.

En étudiant cette logique linéaire, et plus particulièrement des espaces de Köthe dans lesquels elle peut être interprétée ([Ehr02]), Ehrhard et Regnier ont défini une version différentiable du λ -calcul ([ER04]). Ils ont montré que la formule de Taylor, qui permet en analyse, d'approximer une fonction différentiable par un polynôme dont les coefficients sont les dérivées de cette

fonction, peut être transposée dans le cadre du λ -calcul ([ER06]). Boudes, He et Pagani ont récemment donné une caractérisation du développement de Taylor d'un λ -terme ([BHP13]). Le λ -calcul avec ressources est une réécriture du λ -calcul différentiable, ce qui donne un calcul assez proche de la version de Boudol étudiée dans les années 90 ([Bou93, GBL99]) ou de Kfoury ([Kfo00]). C'est un calcul qui contient des ressources dupliquables et des ressources non-dupliquables.

La version du λ -calcul avec ressources considérée ici et celle où toutes les ressources sont non-dupliquables. On peut la voir comme une variante complètement linéaire du λ -calcul. Tout d'abord la syntaxe diffère quelque peu de celle du λ -calcul : l'application ne se fait plus entre deux λ -termes mais entre un λ -terme avec ressources et un multi-ensemble de λ -termes avec ressources. De plus, contrairement à la β -réduction du λ -calcul, la réduction définie dans le λ -calcul avec ressources est une relation entre ensembles de termes. Enfin cette variante ne permet pas de supprimer une ressource ou de la dupliquer, ce qui peut être fait avec le λ -calcul. Dans toute cette thèse on n'étudiera que les termes qui sont uniformes.

D'un certain point de vue, les λ -termes avec ressources sont aux λ -termes purs ce que les polynômes sont aux fonctions réelles.

Le lien qui existe entre le λ -calcul et le λ -calcul avec ressources est le même que celui qu'il y a entre la logique classique et la logique linéaire : il permet d'étudier la façon dont les λ -termes (les éléments du λ -calcul) gèrent leurs ressources. On peut espérer relier ce calcul linéaire avec le récent développement de l'informatique quantique. En effet, en informatique quantique on manipule des qubits (à la place des bits en informatique classique), or la mécanique quantique ne permet pas de dupliquer un état quantique inconnu comme on peut dupliquer un bit sans connaître sa valeur. Ce qui implique qu'on ne peut pas utiliser un programme usuel sur un ordinateur quantique. C'est aussi le cas pour le λ -calcul avec ressources : lors de la réduction on ne peut ni effacer, ni dupliquer un argument. En connaissant mieux le λ -calcul avec ressources on pourrait peut être en déduire un langage de programmation quantique efficace.

Dans un premier temps nous présenterons le λ -calcul avec ressources et ses particularités. Nous définirons ensuite une nouvelle réduction qui a la propriété de préserver l'uniformité des λ -termes avec ressources et qui correspond à la β -réduction. À l'aide de cette nouvelle réduction nous pourrions alors généraliser un des résultats principaux de [ER06]. Puis nous définirons la β -expansion.

Dans un deuxième temps nous étudierons les liens qui existent entre les

termes qui sont issus d'une même réduction. Nous donnerons quelques résultats en fonction de la forme de ces termes, puis nous exposerons une liste d'exemples de termes qui sont issus d'une même réduction et dont les caractéristiques sont intéressantes.

Dans un troisième temps nous détaillerons le programme qui a été écrit dans le cadre de cette thèse pour démontrer le résultat principal §3.2.18. Nous expliquerons les différentes fonctions que l'on a programmées et nous expliquerons pourquoi nous l'avons fait comme cela.

Enfin nous concluerons sur les questions que nous nous sommes posées au cours de ce travail, mais auxquelles nous n'avons pas pu répondre, et sur les problématiques que nous pourrions étudier désormais.

Chapitre 2

Préliminaires

2.1 Notations

Les conventions suivantes seront utilisées dans la suite :

- Les variables seront dénommées par x, y, z .
- Les variables apparaissant dans un terme t seront notées $\mathcal{V}(t)$.
- les variables libres apparaissant dans un terme t seront notées $\mathcal{V}_L(t)$.
- Les termes simples seront dénommés par s, t, u, v, w en minuscule.
- Les monômes de termes seront dénommés par T, U, V, S, W en majuscule.
- Les ensembles seront dénommés par A, B, C .
- On notera les monômes de termes de la façon suivante : $T = [t_1 \dots t_n]$.
- L'ensemble des termes simples sera noté Δ , celui des monômes sera noté $\Delta^!$ et la réunion des deux sera notée $\Delta^{(!)}$.
- On utilisera la notation $\mathcal{M}_{fini}(A)$ pour désigner un multi-ensemble fini d'éléments de A .
- On notera $\downarrow s$ pour désigner l'ensemble des éléments pouvant être obtenus en réduisant s .
- On utilisera les lettres grecques, α, β, γ , pour désigner des adresses.
- Le symbole \star désignera une variable libre.

2.2 Multi-ensembles

§2.2.1 **Définition (Multi-ensemble).** Soit \mathcal{A} un ensemble. Une fonction $S : \mathcal{A} \mapsto \mathbb{N}$ à support fini est appelé un multi-ensemble fini sur \mathcal{A} . Pour $x \in \mathcal{A}$, $S(x)$ est appelé la multiplicité de x . On notera par $|S| = \sum_{x \in \mathcal{A}} S(x)$ le cardinal du multi-ensemble S . Soit S, T deux multi-ensembles sur \mathcal{A} , on définit l'union

de S et T par le multi-ensemble U sur \mathcal{A} tel que $\forall x \in \mathcal{A}, U(x) = S(x) + T(x)$. On utilisera une notation multiplicative $U = S.T$. Le multi-ensemble vide est le multi-ensemble V dont le support est \emptyset , autrement dit tel que pour tout $x \in \mathcal{A}, V(x) = 0$. Il est alors naturel de le noter 1 puisqu'on a $S = 1.S$ pour tout multi-ensemble S . Par abus de langage on identifiera parfois le multi-ensemble $x \mapsto 1$ avec l'élément x .

§2.2.2 **Définition (Monômes).** Soit \mathcal{A} un ensemble, on appelle monôme sur \mathcal{A} un multi-ensemble fini sur \mathcal{A} noté multiplicativement. C'est à dire que pour une famille $(x_i)_{i \leq n}$ d'éléments de \mathcal{A} on notera par $[x_1^{p_1} \dots x_n^{p_n}]$ le multi-ensemble où x_i a la multiplicité p_i et tous les autres éléments de \mathcal{A} ont la multiplicité 0 . Soit S et T deux monômes, on notera simplement par ST l'union de S et T . Naturellement on notera 1 pour le multi-ensemble vide. Par abus de langage on identifiera parfois le monôme $[x^1]$ avec x et on omettra les crochets lorsqu'ils ne sont pas nécessaires, c'est à dire lorsque le support du monôme est un singleton. Par la suite on notera $t \in T$ pour indiquer que t appartient au support de T .

2.3 Le λ -calcul pur

On présentera dans ce chapitre les bases du λ -calcul ainsi que les définitions dont nous auront besoin par la suite. Pour approfondir ce sujet on pourra se référer à [Kri90] et [Bar84].

La syntaxe du λ -calcul est la suivante :

$$M, N ::= x \mid \lambda x M \mid (M)N.$$

On notera Λ l'ensemble des λ -termes.

§2.3.1 **Définition.** Un contexte est un terme avec des trous. On le définit par induction.

- x est un contexte ;
- $[\]$ est un contexte ;
- $\lambda x C[\]$ est un contexte si $C[\]$ est un contexte ;
- $(C[\])C'[\]$ est un contexte si $C[\]$ et $C'[\]$ sont des contextes.

Si $C[\]$ est un contexte et u un λ -terme, alors on note $C[u]$ le résultat du positionnement du terme u dans les trous de $C[\]$. Par cette action on peut éventuellement lier des variables de u (contrairement à la substitution qui est définie ci-après).

§2.3.2 **Exemple.** Soit le contexte $C[\] = \lambda x([\])(x)[\]$ et le terme $u = \lambda y(y)x$. Le terme $C[u]$ est égal à $\lambda x(\lambda y(y)x)(x)\lambda y(y)x$.

§2.3.3 **Définition.** On définit la substitution d'une variable x par un λ -terme t dans un λ -terme u par induction sur u . On notera cette substitution $u[t/x]$, qui se lit « u dans lequel t remplace x ».

- Soit u une variable, alors si $u = x$, $u[t/x] = t$ et si $u \neq x$, $u[t/x] = u$;
- $(\lambda y u_1)[t/x] = \lambda y u_1[t/x]$ si $y \notin \mathcal{V}(t)$ et $(\lambda x u_1)[t/x] = \lambda x u_1$;
- $((u_1)u_2)[t/x] = (u_1[t/x])u_2[t/x]$.

On appelle redex un terme de la forme $(\lambda x u)v$. On introduit une réduction sur ces termes qu'on appelle β -réduction. Le redex $(\lambda x u)v$ se β -réduit en $u[v/x]$ (on note $(\lambda x u)v \beta u[v/x]$).

Cette réduction passe au contexte, c'est à dire que pour tout contexte $C[\]$, on a $C[(\lambda x u)v] \beta C[u[v/x]]$.

De plus le λ -calcul vérifie la propriété du losange. C'est à dire que si un λ -terme t se réduit (en un nombre d'étapes quelconque) en u par une certaine stratégie et en v par une autre, alors il existe un λ -terme w qui est à la fois un réduit de u et de v .

§2.3.4 **Définition.** Un λ -terme est en *forme normale* si il ne contient pas de redex.

§2.3.5 **Définition.** Deux λ -termes sont α -équivalents si ils ne diffèrent que par le nom de leurs variables liées. On définit cette notion par induction.

- x est α -équivalent à x .
- $\lambda x u$ est α -équivalent à $\lambda y (u[y/x])$, si $y \notin \mathcal{V}(u)$.
- $(u)v$ est α -équivalent à $(u')v'$ si u est α -équivalent à u' et v α -équivalent à v' .

On considèrera par la suite l'ensemble des λ -termes quotienté par la relation d' α -équivalence.

§2.3.6 **Définition (Adresse).** Une *adresse* est un mot fini sur l'alphabet $\{a, b, c\}$.

§2.3.7 **Définition.** Soit u un λ -terme et ϵ le mot vide, $\text{Sub}(u, \epsilon)$ est l'ensemble des sous-termes de u et de leur adresse dans u . Si α est l'adresse de u dans un terme t alors on définit par induction $\text{Sub}(u, \alpha)$ qui est l'ensemble des sous-termes de u (et donc de t) et de leur adresse dans t .

- $\text{Sub}(x, \alpha) = \{(x, \alpha)\}$;
- $\text{Sub}(\lambda x v, \alpha) = \{(\lambda x v, \alpha)\} \cup \text{Sub}(v, c\alpha)$;
- $\text{Sub}((v)w, \alpha) = \{((v)w, \alpha)\} \cup \text{Sub}(v, a\alpha) \cup \text{Sub}(w, b\alpha)$.

§2.3.8 **Exemple.** Soit $u = \lambda x(x)y$. On a

$$\text{Sub}(u, \epsilon) = \{(\lambda x(x)y, \epsilon), ((x)y, c), (y, cb), (x, ca)\}.$$

De plus si u est à l'adresse aac dans le λ -terme t alors

$$\text{Sub}(u, aac) = \{(\lambda x(x)y, aac), ((x)y, aacc), (y, aaccb), (x, aacca)\}.$$

§2.3.9 **Définition (Sous-terme).** On dit que u est un sous-terme de t si il existe β tel que (u, β) appartient à $\text{Sub}(t)$, et on notera $u \sqsubset_{\alpha} t$, ou si il existe un contexte $C[]$ tel que $t = C[u]$.

§2.3.10 **Définition.** On dit que α est une adresse active de t si il existe un redex r tel que $r \sqsubset_{\alpha} t$.

On introduit la β -réduction étiquetée, qui permet de réduire un redex précis dans un terme en donnant son adresse.

§2.3.11 **Définition.** Soit u un terme et α une adresse active de u . On notera $u \beta_{\alpha} u'$ pour indiquer que u' est obtenu en réduisant le redex à l'adresse α dans u .

Chapitre 3

Le λ -calcul avec ressources

Le λ -calcul avec ressources est un dérivé du λ -calcul. La différence réside dans le fait qu'il comporte une partie linéaire et non déterministe. Linéaire, c'est à dire qu'on ne s'autorise pas de duplication, ni d'effacement de ressources. Non déterministe car pour réduire un redex il faut faire un choix. Dans toute sa généralité, ce calcul a des ressources linéaires (i.e non duplicables et non effaçables) et des ressources non linéaires ([PT09, PRDR10]). De plus les arguments des applications ne sont plus des termes, mais des monômes de termes, notés multiplicativement.

La version considérée pour la suite est la version de [ER06] légèrement modifiée pour ne regarder que le coté qualitatif et pas quantitatif de ce calcul. C'est celle qui est présentée dans [Ehr10]. Elle est complètement linéaire, c'est à dire que toutes les ressources sont linéaires. De plus on ne considèrera que des termes qui sont uniformes. Toutefois il peut arriver qu'en les réduisant ils perdent cette propriété.

On adopte une syntaxe légèrement différente de celle du λ -calcul pour préciser la linéarité et on distingue les termes simples des arguments qui sont des monômes.

$$\text{Termes simples : } s ::= x \mid \lambda x s \mid \langle s \rangle T$$
$$\text{Monômes : } T ::= 1 \mid [s.T]$$

La linéarité de ce calcul se trouve dans la substitution. On se munit d'un opérateur permettant de substituer une seule occurrence de variable. Ainsi substituer une variable par un monôme de termes correspond à substituer chaque occurrence par un terme du monôme. Comme les monômes ne sont

pas ordonnés il y a un choix à faire pour la substitution, d'où le non déterminisme. On appellera par la suite monôme un monôme de termes.

On a une notion de variable libre et de variable liée. La variable x dans le terme $\lambda x s$ est dite *liée* par le lambda qui précède s . Les autres variables sont dites *libres*. Comme dans le λ -calcul pur.

On adopte la convention de Barendregt, qui stipule que dans un terme les variables libres sont différentes des variables liées. De plus l'égalité entres termes s'entend à α -conversion près.

Définissons maintenant quelques outils dont nous aurons besoin par la suite.

§3.0.12 **Définition (degré).** Le degré d'une variable x dans un terme t est le nombre d'occurrences de x présentent dans t . On le note $deg_x(t)$ et on le définit par induction.

- $deg_x(x) = 1$ et $deg_x(y) = 0$;
- $deg_x(\lambda x s) = 0$ et $deg_x(\lambda y s) = deg_x(s)$;
- $deg_x(\langle s \rangle [t_1 \dots t_n]) = deg_x(s) + deg_x(t_1) + \dots + deg_x(t_n)$.

On se donne également une action qui permet de faire agir les constructeurs du λ -calcul avec ressources sur les ensembles. Soit A un ensemble de termes, B un ensemble de monômes, x une variable, s un terme simple et T un monôme, l'action est la suivante :

$$\begin{aligned} \lambda x A &= \{\lambda x a \mid a \in A\} \\ \langle A \rangle T &= \{\langle a \rangle T \mid a \in A\} \\ \langle s \rangle A &= \{\langle s \rangle a \mid a \in A\} \\ \langle A \rangle B &= \{\langle a \rangle b \mid a \in A, b \in B\} \end{aligned}$$

On a vu dans le chapitre précédent la définition d'un contexte dans le λ -calcul. On va alors définir une notion de contexte avec multiplicités qu'on va appeler contexte linéaire.

§3.0.13 **Définition.** Un *contexte linéaire* $C[]_1 \dots []_n$ est un terme avec n trous. On le définit par induction.

- x est un contexte linéaire ;
- $[]$ est un contexte linéaire ;
- Si $C[]_1 \dots []_n$ est un contexte linéaire alors $\lambda x C[]_1 \dots []_n$ en est un aussi ;
- Si $C[] \dots []$ et $C_1[] \dots []$, \dots , $C_m[] \dots []$ sont des contextes linéaires alors

$$\langle C[] \dots [] \rangle [C_1[] \dots [] \dots C_m[] \dots []]$$

est un contexte que l'on notera $C'[] \dots []$. Pour ne pas surcharger les notations on a volontairement omis la numérotation des trous dans le dernier cas. Les trous du contexte $C'[] \dots []$ sont, dans l'ordre, ceux de C, C_1, \dots, C_m .

D'autre part, si $C[]_1 \dots []_n$ est un contexte linéaire et t_1, \dots, t_n des termes alors $C[t_1]_1 \dots [t_n]_n$ est le résultat de la substitution élémentaire des t_i dans les i -ème trous de $C[]_1 \dots []_n$: en faisant cette substitution on peut lier les variables des t_i .

§3.0.14 **Exemple.** Soit $C[]_1[]_2 = \langle \lambda x \langle x \rangle []_1 \rangle \langle y \rangle 1. \langle y \rangle []_2$ le contexte linéaire et $t_1 = x, t_2 = \star$, alors $C[t_1]_1[t_2]_2 = \langle \lambda x \langle x \rangle x \rangle \langle y \rangle 1. \langle y \rangle \star$.

Remarque : L'égalité entre contextes linéaires est l'égalité littérale, par exemple $\lambda x[] \neq \lambda y[]$.

3.1 β -réduction

Contrairement à la version de [ER06], qui étudie l'aspect quantitatif de la réduction, nous nous intéresserons ici uniquement à l'aspect qualitatif. Ce qui revient à travailler dans le semi-anneau $\{0, 1\} = F_2$. C'est la version du λ -calcul avec ressources qui est présentée dans [Ehr10]. À l'origine on a utilisé la somme pour faire apparaître le non déterminisme de la réduction. Un terme t pouvait alors se réduire en $2u + v$. On pouvait alors, au cours de la réduction, compter le nombre d'occurrences de termes qui apparaissaient. C'est ce qui est fait dans [ER06]. Ici, nous ne voulons pas compter des termes mais plutôt savoir quels termes apparaissent dans la réduction. Pour cela nous avons opté, tout comme dans [Ehr10], pour une approche ensembliste. Nous avons donc remplacé la somme par l'union des ensembles. Nous ferons parfois l'amalgame entre $u + v$ et $\{u, v\}$. De plus, par abus de langage, on notera souvent u pour $\{u\}$.

§3.1.1 **Définition (substitution petit pas).** La substitution petit pas est la substitution d'une occurrence de variable. On la définit par induction et à α -conversion près.

- $x[u/y] = \{u\}$ si $x = y$, \emptyset sinon ;
- $(\lambda x s)[u/y] = \lambda x(s[u/y])$ si $x \notin \mathcal{V}(u)$ et $(\lambda y s)[u/y] = \emptyset$;
- $(\langle s \rangle T)[u/y] = \langle s[u/y] \rangle T \cup \langle s \rangle (T[u/y])$;

Par extension on le définit aussi pour les monômes.

- $([t_1 \dots t_n])[u/y] = \bigcup_{i=1}^n \{[t_1 \dots t_i[u/y] \dots t_n]\}$;

Cette substitution permet d'obtenir toutes les façons de substituer une occurrence de variable dans t par un terme u .

§3.1.2 **Exemple.** $(\lambda x \langle \langle y \rangle x \rangle y^2)[u/y] = \{\lambda x \langle \langle u \rangle x \rangle y^2, \lambda x \langle \langle y \rangle x \rangle [u.y]\}$

Sous certaines conditions, la substitution petit pas commute avec elle même.

§3.1.3 **Proposition.** *Si s, u_1, u_2 sont des termes et $x \notin \mathcal{V}(u_1) \cup \mathcal{V}(u_2)$, alors*

$$(s[u_1/x])[u_2/x] = (s[u_2/x])[u_1/x]$$

Démonstration. On le prouve par induction sur s .

- Si $s = x$, alors $(s[t_1/x])[t_2/x] = (s[t_2/x])[t_1/x] = \emptyset$ car $x \notin \mathcal{V}(t_1) \cup \mathcal{V}(t_2)$. Si $s \neq x$ alors le résultat est évident puisque pour tout terme u , on a $s[u/x] = s$;
- Si $s = \lambda y s$, alors $((\lambda y s)[t_1/x])[t_2/x] = (\lambda y s[t_1/x])[t_2/x] = \lambda y (s[t_1/x])[t_2/x]$ et de même $((\lambda y s)[t_2/x])[t_1/x] = \lambda y (s[t_2/x])[t_1/x]$. Par hypothèse d'induction on a le résultat attendu;
- Si $s = \langle s \rangle T$, alors

$$\begin{aligned} (\langle \langle s \rangle T \rangle [t_1/x])[t_2/x] &= \{\langle s[t_1/x] \rangle T, \langle s \rangle T[t_1/x]\} [t_2/x] \\ &= \{\langle \langle s[t_1/x] \rangle T \rangle [t_2/x], \langle \langle s \rangle T[t_1/x] \rangle [t_2/x]\} \\ &= \{\langle \langle s[t_1/x] \rangle [t_2/x] \rangle T, \langle s[t_1/x] \rangle T[t_2/x] \\ &\quad , \langle s[t_2/x] \rangle T[t_1/x], \langle s \rangle (T[t_1/x])[t_2/x]\}; \end{aligned}$$

De même :

$$\begin{aligned} (\langle \langle s \rangle T \rangle [t_2/x])[t_1/x] &= \{\langle \langle s[t_2/x] \rangle [t_1/x] \rangle T, \langle s[t_2/x] \rangle T[t_1/x] \\ &\quad , \langle s[t_1/x] \rangle T[t_2/x], \langle s \rangle (T[t_2/x])[t_1/x]\}. \end{aligned}$$

Par hypothèse d'induction on a $(s[t_2/x])[t_1/x] = (s[t_1/x])[t_2/x]$ et $(T[t_2/x])[t_1/x] = (T[t_1/x])[t_2/x]$, d'où le résultat. \square

On définit ensuite une substitution qui permet de remplacer toutes les occurrences d'une variable considérée.

§3.1.4 **Définition (substitution grand pas).** La substitution grand pas est la substitution d'une variable par un multi-ensemble de termes. On la définit à partir de la substitution petit pas.

$$\begin{aligned} s[T/x] &= s[t_1/x] \dots [t_n/x] \\ s[1/x] &= \begin{cases} \emptyset & \text{si } x \in \mathcal{V}(s) \\ \{s\} & \text{sinon} \end{cases} \end{aligned}$$

§3.1.5 **Définition.** Soit s un terme (éventuellement un monôme) et T un monôme, on note $\partial_x(s, T)$ le terme s dans lequel toutes les occurrences de x ont été substituées par les éléments de T . On a $\partial_x(s, T) = s[T/x][1/x]$.

§3.1.6 **Théorème.** Soit x une variable, s un terme simple et T un monôme.

$$\partial_x(s, T) \neq \emptyset \text{ si et seulement si } \text{deg}_x(s) = |T|.$$

Démonstration. Montrons d'abord l'implication par contraposée. Supposons que $\text{deg}_x(s) = n$ et $|T| = k$ avec $n \neq k$. Il y a deux possibilités. Soit $n > k$ et alors lorsqu'on a fait la substitution de tous les éléments de T , il reste des occurrences de x dans s . On peut supposer, sans perte de généralité, que $n = k + 1$ et on a

$$\partial_x(s, T) = s[t_1/x] \dots [t_n/x][1/x] = s'[1/x]$$

avec s' qui contient encore une occurrence de x . Par définition de la substitution grand pas on a $\partial_x(s, T) = \emptyset$. Soit $n < k$ et alors lorsqu'on a substitué toutes les occurrences de la variable x dans s il reste des termes dans T . On peut supposer, sans perte de généralité, que $k = n + 1$ et on a

$$\partial_x(s, T) = s[t_1/x] \dots [t_n/x][1/x] = s'[t_n/x][1/x]$$

avec s' qui ne contient plus d'occurrences de x . Donc par définition de la substitution petit pas on a $\partial_x(s, T) = \emptyset$.

Montrons alors la réciproque. On suppose que $\text{deg}_x(s) = |T| = n$. On a

$$\partial_x(s, T) = s[t_1/x] \dots [t_n/x][1/x] = s'[1/x]$$

avec s' qui ne contient plus d'occurrences de la variable x . On a alors $\partial_x(s, T) = s'$ et s' est un λ -terme avec ressources donc différent de \emptyset . □

La notion de réduction est alors la suivante, le redex $\langle \lambda x s \rangle T$ se β -réduit en $\partial_x(s, T)$ et on notera $\langle \lambda x s \rangle T \beta_{\Delta}^1 \partial_x(s, T)$.

La réduction s'étend au contexte, c'est à dire si t' est un réduct de t alors $C[t']$ est un réduct de $C[t]$.

§3.1.7 **Définition (Forme normale).** Soit $t \in \Delta$. On dit que t est en forme normale, si t ne contient pas de redex. Pour $u \in \Delta$, on notera $\text{NF}(u)$ l'ensemble des termes en forme normale pouvant être obtenus par réduction à partir de u .

Une partie du travail de cette thèse consistera à essayer de caractériser $\text{NF}(u)$.

Comme on l'a dit précédemment, le λ -calcul avec ressources est une variante du λ -calcul pur où le nombre d'utilisations des ressources (arguments) est fini. Dans ce sens on peut voir certains de ces termes comme des approximations linéaires des λ -termes purs. Pour cette même raison toute réduction termine, en effet la taille des termes décroît strictement au cours de la réduction.

Pour un redex donné $(\lambda x u)v$, l'argument, v , est utilisé un certain nombre de fois au cours de la réduction. Cette notion n'apparaît pas explicitement dans le λ -calcul pur, mais le λ -calcul avec ressources le permet. Par définition de celui-ci, l'argument d'une application n'est plus un terme mais un monôme de termes, ce qui impose de choisir le nombre d'utilisations de l'argument de chaque application.

§3.1.8 **Définition.** Les approximatants linéaires d'un λ -terme pur sont des λ -termes avec ressources qui lui *ressemblent* :

- $x \in \Delta$ ressemble à $x \in \Lambda$;
- $\lambda x s \in \Delta$ ressemble à $\lambda x u \in \Lambda$, si $s \in \Delta$ ressemble à $u \in \Lambda$;
- $\langle s \rangle [t_1 \dots t_n]$ ressemble à $(u)v$, si $s \in \Delta$ ressemble à $u \in \Lambda$ et pour tout i , $t_i \in \Delta$ ressemble à $v \in \Lambda$;

Les termes qui ressemblent à u sont appelés approximatants linéaires de u .

§3.1.9 **Exemple.** — $\langle x \rangle y$, $\langle x \rangle y^3$ et $\langle x \rangle 1$ ressemblent à $(x)y$

- $\lambda z \langle x \rangle z^3$ ressemble à $\lambda z (x)z$
- $\langle \lambda z \langle x \rangle z^2 \rangle [\langle x \rangle y. \langle x \rangle y^3. \langle x \rangle 1]$ ressemble à $(\lambda z(x)z)(x)y$

On introduit alors une fonction $\mathcal{T}(u)$ qui calcule l'ensemble des termes de Δ qui ressemblent à $u \in \Lambda$.

§3.1.10 **Définition.** $\mathcal{T}(u)$ est défini par induction de la façon suivante :

- $\mathcal{T}(x) = \{x\}$;
- $\mathcal{T}(\lambda x u) = \{\lambda x t \mid t \in \mathcal{T}(u)\}$;
- $\mathcal{T}((u)v) = \{\langle s \rangle T \mid s \in \mathcal{T}(u), T \in \mathcal{M}_{\text{fini}}(\mathcal{T}(v))\}$;

On introduit également une notion de *cohérence* qui permet de caractériser les éléments de $\mathcal{T}(u)$.

§3.1.11 **Définition (Cohérence).** La cohérence est une relation entre λ -termes avec ressources, ou entre monômes, qui est définie par induction :

- $x \circ x$;
- $\lambda x u \circ t'$ si $t' = \lambda x u'$ avec $u \circ u'$;
- $\langle s \rangle T \circ t'$ si $t' = \langle s' \rangle T'$ avec $s \circ s'$ et $T \circ T'$;
- $T \circ T'$ si $\forall t_1 \in T, \forall t_2 \in T', t_1 \circ t_2$;

Remarque : Le monôme vide est noté 1 et il est cohérent avec tous les autres monômes. On utilisera aussi la cohérence entre ensembles qui est la même que celle pour les monômes. On peut voir un ensemble comme un multi-ensemble avec toutes les multiplicités non nulles égales à 1.

§3.1.12 **Proposition.** *La cohérence est une relation symétrique mais non réflexive et non transitive.*

Démonstration. Elle n'est pas réflexive car le monôme xy n'est pas cohérent avec lui-même si $x \neq y$. Elle n'est pas transitive car $x \circ 1$ et $1 \circ y$ mais $x \not\circ y$. \square

§3.1.13 **Définition (clique).** Une clique dans (E, R) , où E est un ensemble et R une relation, est un ensemble d'éléments de E tel que pour tout $e_1, e_2 \in E$ on ait $e_1 R e_2$.

§3.1.14 **Définition.** Une clique est *maximale* si tout élément cohérent avec chaque élément de la clique est dans la clique.

§3.1.15 **Proposition.** *Si $u \in \Lambda$, $\mathcal{T}(u)$ est une clique maximale dans $(\Delta_{\mathcal{R}}, \circ)$.*

Démonstration. On doit montrer que $\forall t_1, t_2 \in \mathcal{T}(u), t_1 \circ t_2$. Montrons le par induction sur u .

- $u = x$ alors par définition $t_1 = x$ et $t_2 = x$ donc $t_1 \circ t_2$;
- $u = \lambda x v$ alors il existe s_1, s_2 tels que $t_1 = \lambda x s_1$ et $t_2 = \lambda x s_2$ avec $s_1, s_2 \in \mathcal{T}(v)$. Par hypothèse d'induction on a $s_1 \circ s_2$, et donc $t_1 \circ t_2$;
- $u = (v)w$ alors il existe s_1, s_2, T_1, T_2 tels que $t_1 = \langle s_1 \rangle T_1$ et $t_2 = \langle s_2 \rangle T_2$ avec $s_1, s_2 \in \mathcal{T}(v)$ et $T_1, T_2 \subset \mathcal{T}(w)$. Par hypothèse d'induction on a $s_1 \circ s_2$ et $T_1 \circ T_2$ puisque tous leurs éléments appartiennent à $\mathcal{T}(w)$. Donc on en déduit que $t_1 \circ t_2$ par définition de la cohérence.

Pour montrer la maximalité on fait une induction sur u . On doit montrer que tout terme cohérent avec chaque élément de $\mathcal{T}(u)$ est dans $\mathcal{T}(u)$.

- Si $u = x$ alors $\mathcal{T}(u) = \{x\}$ donc le résultat est immédiat puisque le seul terme cohérent avec x est lui-même ;
- Si $u = \lambda x v$ alors $\mathcal{T}(u) = \{\lambda x v' \mid v' \in \mathcal{T}(v)\}$. Soit t un terme cohérent avec chaque élément de $\mathcal{T}(u)$ alors t est de la forme $\lambda x s$ tel que s est

cohérent avec chaque v' . Par hypothèse d'induction $s \in \mathcal{T}(v)$. Donc $t \in \mathcal{T}(u)$;

- Si $u = (v)w$ alors $\mathcal{T}(u) = \{\langle v' \rangle W' \mid v' \in \mathcal{T}(v), W' \in \mathcal{M}_{fini}(\mathcal{T}(w))\}$. Soit t un terme cohérent avec chaque élément de $\mathcal{T}(u)$ alors t est de la forme $\langle s \rangle T$ avec s qui est cohérent avec chaque élément de $\mathcal{T}(v)$ et T qui est cohérent avec chaque monôme de $\mathcal{T}(w)$. Par définition de la cohérence on a pour tout $t_i \in T$, t_i est cohérent avec chaque élément de $\mathcal{T}(w)$. Par hypothèse d'induction on en déduit que $s \in \mathcal{T}(v)$ et pour tout $t_i \in T, t_i \in \mathcal{T}(w)$, donc $T \in \mathcal{M}_{fini}(w)$ et donc par définition $t \in \mathcal{T}(u)$.

□

§3.1.16 **Définition.** On dit qu'un terme est *uniforme*, s'il est cohérent avec lui-même.

Remarque : en particulier les éléments de $\mathcal{T}(u)$ sont uniformes pour tout $u \in \Lambda$.

§3.1.17 **Proposition.** Si T est un monôme uniforme alors il existe $u \in \Lambda$ tel que $T \subset \mathcal{T}(u)$.

Démonstration. Le cas où T est vide est trivial. Soit $T = t_1 \dots t_n$, on montre le résultat par induction sur t_1 .

- Si $t_1 = x$ alors par définition de la cohérence on a : $\forall i \leq n, t_i = x$, donc $u = x$ convient ;
- Si $t_1 = \lambda x s_1$ alors par définition de la cohérence on a : $\forall i \leq n, t_i = \lambda x s_i$ et $\forall i, j \leq n, s_i \supset s_j$ donc $S = s_1 \dots s_n \supset S$. Par hypothèse d'induction il existe u' tel que $S \subset \mathcal{T}(u')$ et donc $u = \lambda x u'$ convient.
- Si $t_1 = \langle s_1 \rangle T_1$ alors par définition de la cohérence on a : $\forall i \leq n, t_i = \langle s_i \rangle T_i$ et $\forall i, j \leq n, s_i \supset s_j$ et $T_i \supset T_j$ donc $S = s_1 \dots s_n \supset S$ et $T' = T_1 \dots T_n \supset T'$. Par hypothèse d'induction il existe u' tel que $S \subset \mathcal{T}(u')$ et v tel que $T' \subset \mathcal{T}(v)$ et donc $u = (u')v$ convient.

□

§3.1.18 **Définition.** La *taille* d'un terme avec ressources est définie par induction :

- $\text{taille}(x) = 1$;
- $\text{taille}(\lambda x s) = 1 + \text{taille}(s)$;
- $\text{taille}(\langle s \rangle T) = 1 + \text{taille}(s) + \text{taille}(T)$;
- $\text{taille}([t_1 \dots t_n]) = \sum_{i=1}^n \text{taille}(t_i)$.

Revenons maintenant à la réduction. Nous allons montrer que si on réduit deux redex qui sont cohérents alors leurs réduits sont aussi cohérents. C'est un résultat important présent dans [ER06].

§3.1.19 **Théorème.** Soit $x \in \mathcal{V}$, $s_1, s_2 \in \Delta^{(!)}$ et $T_1, T_2 \in \Delta^!$.

$$\text{Si } s_1 \supset s_2 \text{ et } T_1 \supset T_2 \text{ alors } (\partial_x(s_1, T_1)) \supset (\partial_x(s_2, T_2)) \quad (3.1)$$

$$\text{De plus si } \langle \lambda x s_1 \rangle T_1 \neq \langle \lambda x s_2 \rangle T_2 \text{ alors } (\partial_x(s_1, T_1)) \cap (\partial_x(s_2, T_2)) = \emptyset \quad (3.2)$$

Démonstration. Montrons le par induction sur s_1 . On suppose que $s_1 \supset s_2$ et $T_1 \supset T_2$.

- $s_1 \in \mathcal{V}$ donc $s_1 = s_2$. Si $\partial_x(s_1, T_1) = \emptyset$ ou $\partial_x(s_2, T_2) = \emptyset$ alors le résultat est immédiat. Dans le cas contraire, si $s_1 = x$ alors $T_1 = t_1, T_2 = t_2$ et $(\partial_x(s_1, t_1)) = \{t_1\}$ et $(\partial_x(s_2, t_2)) = \{t_2\}$. Or $T_1 \supset T_2$ donc $t_1 \supset t_2$ et par conséquent $(\partial_x(s_1, t_1)) \supset (\partial_x(s_2, t_2))$. De plus si $\langle \lambda x s_1 \rangle T_1 \neq \langle \lambda x s_2 \rangle T_2$ alors $T_1 \neq T_2$ car $s_1 = s_2$ et donc $(\partial_x(s_1, T_1)) \cap (\partial_x(s_2, T_2)) = \emptyset$.
Si $s_1 \neq x$ alors $T_1 = T_2 = 1$ donc $(\partial_x(s_1, T_1)) = \{s_1\}$ et $(\partial_x(s_2, T_2)) = \{s_2\}$. Or $s_1 \supset s_2$ d'où le résultat. Dans ce cas l'implication 3.2 est triviale puisque $\langle \lambda x s_1 \rangle T_1 \neq \langle \lambda x s_2 \rangle T_2$ n'est jamais vérifié.
- $s_1 = \lambda y u_1$ donc $s_2 = \lambda y u_2$ avec $u_1 \supset u_2$. Par hypothèse d'induction on a le résultat.
- $s_1 = \langle u_1 \rangle U_1$ donc $s_2 = \langle u_2 \rangle U_2$ avec $u_1 \supset u_2$ et $U_1 \supset U_2$. Soit $w_1 \in (\partial_x(s_1, T_1))$, alors il existe v_1 un terme simple et V_1, T_{11}, T_{12} des monômes tels que :

$$\left\{ \begin{array}{l} T_1 = T_{11}T_{12} \\ v_1 \in \partial_x(u_1, T_{11}) \\ V_1 \in \partial_x(U_1, T_{12}) \\ w_1 = \langle v_1 \rangle V_1 \end{array} \right.$$

Soit $w_2 \in (\partial_x(s_2, T_2))$, alors il existe v_2 un terme simple et V_2, T_{21}, T_{22} des monômes tels que :

$$\left\{ \begin{array}{l} T_2 = T_{21}T_{22} \\ v_2 \in \partial_x(u_2, T_{21}) \\ V_2 \in \partial_x(U_2, T_{22}) \\ w_2 = \langle v_2 \rangle V_2 \end{array} \right.$$

De plus comme $T_1 \supset T_2$ alors pour $i, j = 1, 2$ on a $T_{1i} \supset T_{2j}$. On a alors par hypothèse d'induction $v_1 \supset v_2$ et $V_1 \supset V_2$, donc $w_1 = \langle v_1 \rangle V_1 \supset w_2 = \langle v_2 \rangle V_2$.

Montrons alors l'implication 3.2 dans ce cas par contraposée.

Soit $w_1 = \langle v_1 \rangle V_1 \in (\partial_x(s_1, T_1)) \cap (\partial_x(s_2, T_2))$. Par hypothèse d'induction on a

$$\left\{ \begin{array}{ll} v_1 \in (\partial_x(u_1, T_{11})) \cap (\partial_x(u_2, T_{21})) & \Rightarrow u_1 = u_2 \quad \text{et } T_{11} = T_{21} \\ V_1 \in (\partial_x(U_1, T_{12})) \cap (\partial_x(U_2, T_{22})) & \Rightarrow U_1 = U_2 \quad \text{et } T_{12} = T_{22} \end{array} \right.$$

$$\implies \begin{cases} T_1 = T_{11}T_{12} & = & T_2 = T_{21}T_{22} \\ s_1 = \langle u_1 \rangle U_1 & = & s_2 = \langle u_2 \rangle U_2 \end{cases}$$

D'où le résultat.

- $s_1 = u_1U_1$ donc $s_2 = u_2U_2$. Soit $w_1 \in (\partial_x(s_1, T_1))$, alors il existe v_1 un terme simple et V_1, T_{11}, T_{12} des multi ensembles tels que :

$$\begin{cases} T_1 = T_{11}T_{12} \\ v_1 \in \partial_x(u_1, T_{11}) \\ V_1 \in \partial_x(U_1, T_{12}) \\ w_1 = v_1V_1 \end{cases}$$

Soit $w_2 \in (\partial_x(s_2, T_2))$, alors il existe v_2 un terme simple et V_2, T_{21}, T_{22} des monômes tels que :

$$\begin{cases} T_2 = T_{21}T_{22} \\ v_2 \in \partial_x(u_2, T_{21}) \\ V_2 \in \partial_x(U_2, T_{22}) \\ w_2 = v_2V_2 \end{cases}$$

De plus comme $T_1 \supset T_2$ alors pour $i, j = 1, 2$ on a $T_{1i} \supset T_{2j}$. On en déduit par hypothèse d'induction que $v_1 \supset v_2$ et $V_1 \supset V_2$, donc $w_1 = v_1V_1 \supset w_2 = v_2V_2$. Montrons alors l'implication 3.2 par contraposée. Soit $w = vV \in (\partial_x(s_1, T_1)) \cap (\partial_x(s_2, T_2))$. Par hypothèse d'induction on obtient

$$\begin{cases} v \in (\partial_x(u_1, T_{11})) \cap (\partial_x(u_2, T_{21})) & \Rightarrow & u_1 = u_2 & \text{et} & T_{11} = T_{21} \\ V \in (\partial_x(U_1, T_{12})) \cap (\partial_x(U_2, T_{22})) & \Rightarrow & U_1 = U_2 & \text{et} & T_{11} = T_{21} \end{cases}$$

$$\implies \begin{cases} T_1 = T_{11}T_{12} & = & T_2 = T_{21}T_{22} \\ s_1 = u_1U_1 & = & s_2 = u_2U_2 \end{cases}$$

D'où le résultat. □

§3.1.20 **Proposition.** *La réduction définie précédemment, β_Δ , est confluente et fortement normalisante.*

Démonstration. La normalisation forte vient de la linéarité. En effet montrons que la linéarité impose la décroissance stricte de la taille d'un terme. Soit n la taille de $\langle \lambda x s \rangle T$, et $t \in \partial_x(s, T)$. Remarquons que si $\partial_x(s, T)$ est non vide alors il est égal à $s[T/x]$. La taille d'un élément de $\partial_x(s, T)$ est alors égale à la taille de s plus la taille de T auquel on soustrait le nombre d'occurrences de x puisque la taille de x est 1. Or n peut se décomposer de

la façon suivante : $n = 1 + n_1 + n_2$ où n_1 est la taille de $\lambda x s$ et n_2 est la taille de T , ce qui est aussi égal à $2 + n'_1 + n_2$ où n'_1 est la taille de s . Soit k la taille de t alors $k = n'_1 + n_2 - \text{deg}_x(t)$ et par conséquent $k < n$. Ce qui assure la normalisation forte.

Montrons maintenant la confluence. Soit t un terme et r_1, r_2 deux redex présents dans ce terme. Montrons que si on réduit r_1 d'un côté pour obtenir σ_1 (un ensemble de termes) et r_2 de l'autre pour obtenir σ_2 alors on peut réduire chacun des termes présents dans σ_1 et σ_2 afin d'obtenir le même ensemble σ . Deux cas sont à étudier.

— Soit r_1 et r_2 sont distincts, et on a deux cas possibles. On peut supposer, sans perdre en généralité, que t est de la forme $\langle r_1 \rangle r_2$ ou alors de la forme $\langle s \rangle [r_1.r_2]$. Considérons le premier cas où r_1 et r_2 ne sont pas dans le même monôme. Si on réduit d'abord r_1 dans t on obtient des termes de la forme $\langle r'_1 \rangle r_2$ avec r'_1 qui appartient aux réduits de r_1 . En réduisant ensuite r_2 on obtient des termes de la forme $\langle r'_1 \rangle r'_2$ avec r'_2 qui appartient aux réduits de r_2 . Par symétrie on montre qu'en réduisant d'abord r_2 puis r_1 on obtient les mêmes termes.

De même pour le cas où les deux redex sont dans le même monôme. Avec les mêmes notations, si on réduit d'abord r_1 on obtient des termes de la forme $\langle s \rangle [r'_1.r_2]$, puis en réduisant r_2 on obtient des termes de la forme $\langle s \rangle [r'_1.r'_2]$. Par symétrie on montre qu'en changeant l'ordre de réduction des redex on obtient les mêmes termes.

— Soit r_1 et r_2 sont imbriqués, c'est à dire que r_1 est un sous-terme de r_2 (ou r_2 est un sous-terme de r_1). Il y a à nouveau deux cas. Soit r_1 est dans les arguments de r_2 et on peut supposer que t est égal à r_2 qui est de la forme $\langle \lambda x s \rangle [r_1.s_2 \dots s_n]$. Si on réduit r_1 on obtient des termes de la forme $\langle \lambda x s \rangle [r'_1.s_2 \dots s_n]$ où r'_1 représente un des réduits de r_1 . On réduit ensuite r_2 , c'est à dire le redex principal, et on obtient $\partial_x(s, [r'_1.s_2 \dots s_n])$. Changeons alors l'ordre de la réduction et commençons par réduire r_2 . On obtient alors $\partial_x(s, [r_1.s_2 \dots s_n])$, puis on réduit r_1 et on aboutit à $\partial_x(s, [r'_1.s_2 \dots s_n])$.

Soit, et c'est le cas le plus intéressant, r_1 est dans la partie fonctionnelle de r_2 . Dans ce cas on peut supposer que t est égal à r_2 qui est de la forme $\langle \lambda x \langle \lambda y s_1 \rangle T_1 \rangle T_2$. Si on réduit d'abord r_1 on obtient $\langle \lambda x s' \rangle T_2$ avec $s' \in \partial_y(s, T_1)$. Puis en réduisant r_2 on obtient $\partial_x(s', T_2)$. Faisons maintenant l'inverse et réduisons d'abord r_2 . On obtient $\partial_x(\langle \lambda y s \rangle T_1, T_2)$. Ce qui se réduit en $s[T_2/x][T_1[T_2/x]]/y$ qui est équivalent à $s[T_1/y][T_2/x]$. Ce dernier est exactement le résultat de la première réduction.

Donc dans tous les cas en réduisant r_1 puis r_2 ou l'inverse on obtient le même

ensemble de termes σ .

□

Remarque : La réduction ne conserve pas l'uniformité.

§3.1.21 **Exemple.** Le terme uniforme $\langle y \rangle [\langle \lambda x \langle x \rangle y \rangle y \cdot \langle \lambda x \langle x \rangle 1 \rangle y]$ se réduit en une étape en $\langle y \rangle [\langle y \rangle y \cdot \langle \lambda x \langle x \rangle 1 \rangle y]$ qui n'est pas uniforme puisque les deux éléments du monôme ne sont pas cohérents.

On a défini dans la partie sur le λ -calcul pur l'ensemble des sous-termes d'un terme donné. On définit ici la notion équivalente qui diffère légèrement de par le fait qu'on définit cette fois-ci le multi-ensemble des sous-termes et qu'on est obligé de préciser quelle adresse on donne aux différents arguments d'une application.

§3.1.22 **Définition.** Soit t un terme et α une adresse. On définit $\text{Sub}(t, \alpha)$ le multi-ensemble des sous-termes de t par induction sur t .

- $\text{Sub}(x, \alpha) = [(x, \alpha)]$
- $\text{Sub}(\lambda x s, \alpha) = [(\lambda x s, \alpha)] \cup \text{Sub}(s, c\alpha)$
- $\text{Sub}(\langle s \rangle T, \alpha) = [(\langle s \rangle T, \alpha)] \cup \text{Sub}(s, a\alpha) \cup \text{Sub}(T, b\alpha)$
- $\text{Sub}(t_1 \dots t_n, \alpha) = \bigcup_{i=1}^n \text{Sub}(t_i, \alpha)$

§3.1.23 **Exemple.** On choisit un exemple similaire à celui donné dans la partie sur le λ -calcul pur. Soit $t = \lambda x \langle x \rangle y^3$. On a alors

$$\text{Sub}(t, \epsilon) = [(\lambda x \langle x \rangle y^3, \epsilon) \cdot (\langle x \rangle y^3, c) \cdot (y, cb)^3 \cdot (x, ca)].$$

On définit alors une notion similaire à la précédente, mais qui calcule les sous-termes libres de t , c'est à dire les sous-termes qui ne contiennent aucune occurrence de variable liée dans t .

§3.1.24 **Définition.** Soit t un terme et α une adresse. On définit le multi-ensemble $\text{Sub}_L(t, \alpha)$ par induction sur t

- $\text{Sub}_L(x, \alpha) = [(x, \alpha)]$
- $\text{Sub}_L(\lambda x s, \alpha) = [(\lambda x s)] \cup \{(u, \beta) \mid (u, \beta) \in \text{Sub}(s, c\alpha) \text{ et } x \notin \mathcal{V}(u)\}$.
- $\text{Sub}_L(\langle s \rangle T, \alpha) = [(\langle s \rangle T, \alpha)] \cup \text{Sub}_L(s, a\alpha) \cup \text{Sub}_L(T, b\alpha)$
- $\text{Sub}_L(t_1 \dots t_n, \alpha) = \bigcup_{i=1}^n \text{Sub}_L(t_i, \alpha)$

§3.1.25 **Exemple.** Avec le même t que précédemment on a $\text{Sub}_L(t, \epsilon) = [(y, cb)^3]$ mais par contre $\text{Sub}_L(\langle x \rangle y^3, c) = [(\langle x \rangle y^3, c) \cdot (y, cb)^3 \cdot (x, ca)] = \text{Sub}(\langle x \rangle y^3, c)$. En effet vue depuis l'adresse c la variable x est libre puisque l'abstraction qui la lie n'apparaît pas. On verra l'importance de cette notion par la suite.

Les adresses de s dans t sont représentées par l'ensemble noté

$Ad_t^s = \{\beta \in \{a, b, c\}^* \mid (s, \beta) \in \text{Sub}(t, \alpha)\}$. On notera $\text{Sub}(t)$ pour $\text{Sub}(t, \epsilon)$.

§3.1.26 **Proposition.** *Soit u et t deux termes. Il existe β une adresse telle que $(u, \beta) \in \text{Sub}(t, \alpha)$ si et seulement si il existe un contexte C tel que $t = C[u]$.*

Démonstration. Démontrons l'implication par induction sur t . On suppose qu'il existe β telle que $(u, \beta) \in \text{Sub}(t, \alpha)$.

- Si $t = x$ alors $u = x$ et le résultat est immédiat ;
- Si $t = \lambda x s$ alors $(u, \beta) \in \text{Sub}(s, a\alpha)$ (le cas où $u = t$ est immédiat). Par hypothèse d'induction on a l'existence d'un contexte C tel que $s = C[u]$ et par définition des contextes on obtient $C' = \lambda x C$ tel que $t = C'[u]$;
- Si $t = \langle s \rangle T$ alors il y a deux cas à distinguer.
Soit $(u, \beta) \in \text{Sub}(s, a\alpha)$, et par hypothèse d'induction il existe C tel que $s = C[u]$ et donc on en déduit $C' = \langle C \rangle T$ tel que $t = C'[u]$.
Soit $(u, \beta) \in \text{Sub}(T, b\alpha)$, si $T = [t_1 \dots t_n]$ alors il existe i tel que $(u, \beta) \in \text{Sub}(t_i, a\alpha)$. Par hypothèse d'induction il existe C tel que $t_i = C[u]$ et donc on en déduit $C' = \langle s \rangle [t_1 \dots C \dots t_n]$ tel que $t = C'[u]$.

On montre également la réciproque par induction sur t . Supposons qu'il existe un contexte C tel que $t = C[u]$.

- Si $t = x$ alors le résultat est immédiat ;
- Si $t = \lambda x s$ alors, si $u \neq t$ il existe C' tel que $s = C'[u]$, sinon c'est immédiat. Par hypothèse d'induction il existe β tel que $(u, \beta) \in \text{Sub}(s, c\alpha)$ et on en déduit $\beta' = c\beta$ tel que $(u, \beta') \in \text{Sub}(t, \alpha)$;
- Si $t = \langle s \rangle T$ alors il y a deux cas à distinguer.
Soit il existe C_1 tel que $s = C_1[u]$ et alors par hypothèse d'induction il existe β tel que $(u, \beta) \in \text{Sub}(s, a\alpha)$ et on en déduit $\beta' = a\beta$ tel que $(u, \beta') \in \text{Sub}(t, \alpha)$.
Soit il existe C_2 et $i \in \{1, \dots, n\}$ tels que $t_i = C_2[u]$. Par hypothèse d'induction il existe β tel que $(u, \beta) \in \text{Sub}(t_i, b\alpha)$ et donc on en déduit $\beta' = b\beta$ tel que $(u, \beta') \in \text{Sub}(t, \alpha)$.

□

§3.1.27 **Définition (Sous-terme).** On dit que u est un sous-terme de t si il existe β tel que (u, β) appartient à $\text{Sub}(t)$, et on notera $u \sqsubset_\beta t$, ou si il existe un contexte linéaire $C[\]$ tel que $t = C[u]$.

§3.1.28 **Définition (Sous-terme libre).** La notion de sous-terme libre est un peu différente. On dit que u est un sous-terme libre de t si il existe β tel que $(u, \beta) \in \text{Sub}_L(t)$, et on notera $u \sqsubset_\alpha^L t$, c'est à dire si u est un sous-terme de t et ne contient pas de variables liées par un lambda en dehors de u dans t .

§3.1.29 **Proposition.** *Soit t, u, v des termes. Il existe α tel que $u, v \sqsubset_\alpha t$ si et seulement si il existe un monôme T apparaissant dans t , des termes $u', v' \in T$ et une adresse β tels que $u \sqsubset_\beta u', v \sqsubset_\beta v'$.*

Démonstration. La réciproque est immédiate par définition des sous-termes. Par contraposée, si un tel monôme n'existe pas alors α ne contient aucun b . En effet si α contenait un b alors par définition des sous-termes u et v appartiendraient bien à un même monôme T . Si l'adresse α ne contient pas de b alors l'ensemble des termes à cette adresse est un singleton. Et donc $u = v$. \square

§3.1.30 **Proposition.** *Soit s, t des termes. On a $s \supset t$ ssi pour tout $\alpha \in \Sigma^*$, pour tout $s', t' \in \Delta$ tels que $s' \sqsubset_\alpha s, t' \sqsubset_\alpha t$ on a $s' \supset t'$.*

Démonstration. On fait la preuve de l'implication par induction sur t .

- si $t = x$ la conclusion est immédiate ;
- si $t = \lambda x u$ alors $s = \lambda x v$ avec $u \supset v$. Par définition de l'adresse, u et v ont la même adresse. Par hypothèse d'induction on a $\forall \alpha \in \Sigma^*, \forall u', v' \in \Delta$ tels que $u' \sqsubset_\alpha u, v' \sqsubset_\alpha v, u' \supset v'$. Or $u' \sqsubset_\alpha u$ implique $u' \sqsubset_{c\alpha} t$ et de même pour v' ;
- si $t = \langle u \rangle U$ alors $s = \langle v \rangle V$ avec $u \supset v$ et $U \supset V$. Pour $\alpha = a$ et $\alpha = b$ c'est vrai et par hypothèse d'induction c'est vrai pour toute adresse de u et de U donc pour toute adresse de t .
- si $t = t_1 \dots t_n$ alors $s = s_1 \dots s_n$ avec pour tout $i, j, t_i \supset s_j$. Comme tous ces termes ont la même adresse, on conclut par hypothèse d'induction.

La réciproque découle de la définition de la cohérence. \square

§3.1.31 **Corollaire.** *Si t est uniforme alors pour toute adresse α , tous les termes d'adresse α dans t sont cohérents et réciproquement.*

3.2 Γ -réduction

On va définir une nouvelle réduction qui a pour but de préserver l'uniformité à chaque étape.

§3.2.1 **Proposition.** *Si t est uniforme alors pour tout α tel que $r \sqsubset_\alpha t$ où r est un redex, tous les termes à l'adresse α sont des redex.*

Démonstration. Ça découle du corollaire précédent. \square

§3.2.2 **Définition.** Soit t un terme. Si α est une adresse et r un redex tel que $r \sqsubset_\alpha t$, alors α est dite *active*.

§3.2.3 **Définition (réduction à pas de géant).** Soit α une adresse active dans t . Alors t se réduit à pas de géant en $\tau \subset_{fini} \Delta$ selon α , et on notera $t \Gamma_\alpha \tau$, en réduisant tous les sous-termes r dans t tels que $r \sqsubset_\alpha t$. Par convention la réduction à pas de géant selon α n'a aucun effet si α n'est pas une adresse active et en particulier si α pointe vers un monôme vide, i.e. $t \Gamma_\alpha t$ si α n'est pas active dans t .

Remarque : Cette réduction passe aussi au contexte, c'est à dire que si t se réduit en un ensemble τ selon une adresse α , alors pour tout $C[]$, $C[t] \Gamma_{\beta\alpha} \{C[u] \mid u \in \tau\}$ où β est telle que $t \sqsubset_\beta C[t]$.

§3.2.4 **Définition.** On étend la cohérence aux contextes de la façon suivante,

$C_1[] \supset C_2[]$ si pour toute clique $\{u_1, \dots, u_{n+k}\}$,
 $C_1[u_1]_1 \dots [u_n]_n \supset C_2[u_{n+1}]_1 \dots [u_{n+k}]_k$.

Remarque : Si x est une variable alors pour tous contextes $C[], C'[]$ qui ne contiennent pas la variable x on a $C[] \supset C'[]$ si $C[x] \supset C'[x]$.

§3.2.5 **Proposition.** On peut définir inductivement la cohérence pour les contextes de la manière suivante (on écrit C pour $C[]_1 \dots []_{n_C}$ pour simplifier l'écriture) :

- $[] \supset []$;
- $x \supset y$ si $x = y$;
- $\lambda x C \supset C'$ si $C' = \lambda x C''$ où $C'' \supset C$;
- $\langle C \rangle [C_1 \dots C_l] \supset C'$ si $C' = \langle C'' \rangle [C''_1 \dots C''_m]$ où $C \supset C''$ et pour tout $i, j, C_i \supset C''_j$.

Démonstration. On fait la preuve par induction sur C .

- Si $C = x$ alors le résultat est immédiat ;
- Si $C = []$ alors pour tout $\{u, v\}$, une clique, $C[u] \supset C'[v]$. Donc $u \supset C'[v]$, et en particulier $x \supset C'[x]$, d'où $C' = []$;
- Si $C = \lambda x C_1$ alors pour toute clique $\{u, v\}$, $C[u] = \lambda x C_1[u] \supset C'[v]$ et donc $C'[v] = \lambda x C'_1[v]$ avec $C_1[u] \supset C'_1[v]$. Par hypothèse d'induction on a $C_1[] \supset C'_1[]$ et donc $C' = \lambda x C'_1[]$.
- Si $C = \langle C_1 \rangle [C_2 \dots C_n]$ alors pour toute clique $\{u_1, \dots, u_{n+l}\}$, $C[u_1]_1 \dots [u_k]_k =$

$$\langle C_1[u_1]_1 \dots [u_{k_1}]_{k_1} \rangle [C_2[u_{k_1+1}]_1 \dots [u_{k_1+k_2}]_{k_2} \dots C_n[u_{k-k_n+1}]_1 \dots [u_k]_{k_n}]$$

et est cohérent avec $C'[u_{n+1}]_1 \dots [u_{n+l}]_l$ donc $C'[u_{n+1}]_1 \dots [u_{n+l}]_l =$

$$\langle C'_1[u_{n+1}]_1 \dots [u_{n+l_1}]_{l_1} \rangle [C'_2[u_{n+l_1+1}]_1 \dots [u_{n+l_1+l_2}]_{l_2} \dots C'_m[u_{n+l-l_m+1}]_1 \dots [u_{n+k}]_{l_m}]$$

avec $C'_1[u_{n+1}]_1 \dots [u_{n+l_1}]_{l_1} \supset C_1[u_1]_1 \dots [u_{k_1}]_{k_1}$ et pour tout $i > 1, j > 1$,

$$C'_j[u_{n+1+\sum_{p=1}^j l_p}]_1 \dots [u_{n+\sum_{p=1}^{j+1} l_p}]_{l_{j+1}} \supset C_j[u_{1+\sum_{p=1}^j k_p}]_1 \dots [u_{\sum_{p=1}^{j+1} k_p}]_{k_{j+1}}.$$

Par hypothèse d'induction on a $C'_1 \supset C_1$ et pour tout $i > 1, j > 1, C'_i \supset C_j$. Donc $C' = \langle C'_1 \rangle [C'_2 \dots C'_m]$.

□

Remarque : Pour montrer des propriétés de cohérence entre contextes, on peut se restreindre à les montrer sur des contextes à un seul trou, car en analysant la définition par induction on constate que le nombre de trous diminue strictement à chaque fois qu'on utilise l'hypothèse d'induction. C'est ce qu'on fera dans la suite.

§3.2.6 **Lemme.** *Si $C[\] \supset C'[\]$ alors il existe une adresse β telle que pour tout terme u , $(u, \beta) \in \text{Sub}(C[u]) \cap \text{Sub}(C'[u])$ et réciproquement.*

Démonstration. On raisonne par induction sur $C[\]$ pour montrer l'implication.

- Si $C[\] = x$ alors par cohérence $C'[\] = x$ est le résultat est immédiat ;
- Si $C[\] = [\]$ alors par cohérence $C'[\] = [\]$. Pour tout terme u l'adresse vide vérifie la condition voulue ;
- Si $C[\] = \lambda x C_1[\]$ alors par cohérence on a $C'[\] = \lambda x C'_1[\]$. Par hypothèse d'induction pour tout terme u il existe une adresse α telle que $(u, \alpha) \in \text{Sub}(C_1[\]) \cap \text{Sub}(C'_1[\])$ et on en déduit que $\beta = c\alpha$ convient ;
- Si $C[\] = \langle C_1[\] \rangle T$ alors par cohérence on a $C'[\] = \langle C'_1[\] \rangle T'$ tel que $C_1[\] \supset C'_1[\]$ et $T \supset T'$. Par hypothèse d'induction on a pour tout terme u il existe une adresse α telle que $(u, \alpha) \in \text{Sub}(C_1[\]) \cap \text{Sub}(C'_1[\])$, on en déduit que $\beta = a\alpha$ convient ;
- Si $C[\] = \langle s \rangle [C_1[\].T]$ alors par cohérence on a $C'[\] = \langle s' \rangle [C'_1[\].T']$ tel que $s \supset s'$ et $C_1[\] \supset C'_1[\]$. Par hypothèse d'induction on a pour tout terme u , il existe α telle que $(u, \alpha) \in \text{Sub}(C_1[\]) \cap \text{Sub}(C'_1[\])$, on en déduit que $\beta = b\alpha$ convient.

La réciproque vient de la définition de la cohérence entre contextes et de §3.1.30. □

On peut maintenant montrer que la réduction que nous venons de définir a bien les propriétés voulues.

§3.2.7 **Proposition.** *Soit s_1, s_2 des termes et α une adresse active de s_1 . Si $s_1 \supset s_2$ alors soit $\sigma, \tau \subset_{\text{fini}} \Delta$ tels que $s_1 \Gamma_\alpha \sigma$ et $s_2 \Gamma_\alpha \tau$; pour tout $t_1 \in \sigma$, $t_2 \in \tau$, $t_1 \supset t_2$.*

Démonstration. Dans le cas où il n'y a pas de terme dont l'adresse est α dans s_2 alors le résultat découle de §3.1.30. En effet on sait que puisque $s_1 \supset s_2$ alors pour tout $s'_1 \sqsubset_\beta s_1$ et $s'_2 \sqsubset_\beta s_2$ on a $s'_1 \supset s'_2$. Si il n'y a pas de terme à l'adresse α c'est que cette adresse pointe vers un monôme vide. Dans ce cas la réduction selon α ne fait rien sur s_2 et comme tous les termes sont cohérents avec le monôme vide alors en particulier les réduits des redex présents à l'adresse α dans s_1 le sont aussi. Sinon α est également une adresse active de s_2 . Alors il existe $r_{11}, \dots, r_{1n}, r_{21}, \dots, r_{2k}$ des redex cohérents deux à deux tels que pour tout i , $r_{1i} \sqsubset_\alpha s_1$ et $r_{2i} \sqsubset_\alpha s_2$. Donc d'après §3.1.26 il existe $C_1[]_1 \dots []_n, C_2[]_1 \dots []_k$ tels que $s_1 = C_1[r_{11}]_1 \dots [r_{1n}]_n$ et $s_2 = C_2[r_{21}] \dots [r_{2k}]$. Soit r'_{ij} un réduct de r_{ij} et s'_1 (resp. s'_2) un terme appartenant aux réduits de s_1 (resp. s_2) tels que pour tout i , $r'_{1i} \sqsubset_\alpha s'_1$ (resp. $r'_{2i} \sqsubset s'_2$). Par définition on a $C_1[]_1 \dots []_n \supset C_2[]_1 \dots []_k$. Par passage au contexte de la réduction on a $s'_1 = C_1[r'_{11}] \dots [r'_{1n}]$ et $s'_2 = C_2[r'_{21}] \dots [r'_{2k}]$. Par définition de la cohérence entre contextes on a alors $C_1[r'_{11}] \dots [r'_{1n}] \supset C_2[r'_{21}] \dots [r'_{2k}]$. \square

§3.2.8 **Corollaire.** *Cette réduction conserve l'uniformité, c'est à dire que si t est uniforme alors ses réduits le sont également.*

On étend ensuite la notion d'approximants linéaires aux contextes.

§3.2.9 **Définition.** On définit les approximants linéaires pour les contextes de la même façon que pour les termes.

- $\mathcal{T}(x) = \{x\}$;
- $\mathcal{T}([]) = \{[]\}$;
- $\mathcal{T}(\lambda x C) = \{\lambda x C' \mid C' \in \mathcal{T}(C)\}$;
- $\mathcal{T}((C_1)C_2) = \{(C'_1)[C'_2 \dots C'_n] \mid C'_1 \in \mathcal{T}(C_1), \forall i > 1, C'_i \in \mathcal{T}(C_2)\}$.

§3.2.10 **Proposition.** *Si $C'[]_1 \dots []_n$ est un approximant linéaire de $C[]$ alors soit $u \in \Lambda$ pour tout s_1, \dots, s_n approximants linéaires de u on a $C'[s_1]_1 \dots [s_n]_n \in \mathcal{T}(C[u])$.*

Démonstration. On raisonne par induction sur C .

- Si $C = x$ alors $C' = x$ et le résultat est immédiat ;
- Si $C = []$ alors $C' = []$ et le résultat est direct puisque pour tout $u \in \Lambda$, $C[u] = u$;
- Si $C = \lambda x C_1[]$ alors par définition $C' = \lambda x C'_1[]_1 \dots []_n$ avec C'_1 un approximant de C_1 . Par hypothèse d'induction on a pour tout

$u \in \Lambda$, pour tout s_1, \dots, s_n approximants de u , $C'_1[s_1]_1 \dots [s_n]_n$ est un approximant de $C_1[u]$. Et par définition $\lambda x s_i$ est un approximant de $\lambda x u$ donc C' est bien un approximant de C ;

- Si $C = (C_1[\])C_2[\]$ alors par définition $C' = \langle C'_1[\]_1 \dots [\]_{n_1} \rangle [C'_2[\]_1 \dots [\]_{n_2} \dots C'_k[\]_1 \dots [\]_{n_k}]$ avec C'_1 un approximant de C_1 et pour tout $i > 2$, C'_i est un approximant de C_i . Par hypothèse d'induction pour tout $u \in \Lambda$ et pour tout s_1, \dots, s_n , où n est le max des n_i , on a les faits suivants :

$$C'_1[s_1]_1 \dots [s_{n_1}]_{n_1} \in \mathcal{T}(C_1[u]) \quad (3.3)$$

$$\forall i > 2, C'_i[s_1]_1 \dots [s_{n_i}]_{n_i} \in \mathcal{T}(C_i[u]). \quad (3.4)$$

Par définition des approximants linéaires on a

$$\langle C'_1[s_1]_1 \dots [s_{n_1}]_{n_1} \rangle [C'_2[s_1]_1 \dots [s_{n_2}]_{n_2} \dots C'_k[s_1]_1 \dots [s_{n_k}]_{n_k}] \in \mathcal{T}((u)u)$$

et donc $C' = \langle C'_1[\]_1 \dots [\]_{n_1} \rangle [C'_2[\]_1 \dots [\]_{n_2} \dots C'_k[\]_1 \dots [\]_{n_k}]$ est bien un approximant de $C = (C_1[\])C_2[\]$. \square

§3.2.11 **Lemme.** Soit u, v des λ -termes purs, s un terme avec ressources et T un monôme tels que $s \in \mathcal{T}(u)$ et $T \in \mathcal{M}_{fini}(\mathcal{T}(v))$. Soit $w \in \partial_x(s, T)$, alors $w \in \mathcal{T}(u[v/x])$.

Démonstration. On raisonne par induction sur u .

- Si $u = x$ le résultat est immédiat ;
- Si $u = \lambda x u'$ alors $s = \lambda x s'$ avec $s' \in \mathcal{T}(u')$. Par hypothèse d'induction, soit $w \in \mathcal{T}(u'[v/x])$ alors $w \in \partial_x(s', T)$. Or $u[v/x] = \lambda x (u'[v/x])$ et $\partial_x(s, T) = \{\lambda x s'' \mid s'' \in \partial_x(s', T)\}$, de plus $\mathcal{T}(u[v/x]) = \{\lambda x t \mid t \in \mathcal{T}(u'[v/x])\}$. On en déduit que $\lambda x w \in \partial_x(s, T) \cap \mathcal{T}(u[v/x])$;
- Si $u = (u')u''$ alors $s = \langle s' \rangle S''$ avec $s' \in \mathcal{T}(u')$ et $S'' \in \mathcal{M}_{fini}(\mathcal{T}(u''))$. On notera $S'' = [s''_1 \dots s''_k]$. Par hypothèse d'induction, soit $w' \in \partial_x(s', T)$ alors $w' \in \mathcal{T}(s'[v/x])$ et pour tout i si $w''_i \in \partial_x(s''_i, T)$ alors $w''_i \in \mathcal{T}(u''[v/x])$. Or $u[v/x] = (u'[v/x])u''[v/x]$ et donc $\mathcal{T}(u[v/x]) = \{\langle t' \rangle T'' \mid t' \in \mathcal{T}(u'[v/x]), T'' \in \mathcal{M}_{fini}(\mathcal{T}(u''[v/x]))\}$, et $\partial_x(s, T) = \bigcup_{T_1 T_2 = T} (\partial_x(s', T_1) \partial_x(S'', T_2))$. On en déduit que $\langle w' \rangle W'' \in \mathcal{T}(u[v/x]) \cap \partial_x(s, T)$. \square

On peut alors montrer que la réduction à pas de géant correspond exactement à la β -réduction du λ -calcul.

§3.2.12 **Proposition.** *Soit $u \in \Lambda, t \in \mathcal{T}(u)$, si $u \beta_\alpha v$ alors pour tout s appartenant aux réduits de t selon α , $s \in \mathcal{T}(v)$.*

Démonstration. Soit $u, r \in \Lambda$ et $C[\]$ tel que $u = C[r]$ avec r un redex tel que $r \sqsubset_\alpha u$. Soit $t \in \mathcal{T}(u)$ alors $t = C'[r_1]_1 \dots [r_n]_n$ tel que C' est un approximant de C et les r_i sont des approximants de r . De plus pour tout i , $r_i \sqsubset_\alpha t$. Soit v le terme obtenu en réduisant r en r' dans u , alors $v = C[r']$. Soit τ la clique de termes telle que $t \Gamma_\alpha \tau$. Soit $s \in \tau$, par passage au contexte de la réduction on a $s = C'[r'_1]_1 \dots [r'_n]_n$ où r'_i appartient aux réduits du redex r_i . On doit montrer que les r'_i sont des approximants de r' . On utilise pour cela le lemme précédent. En appliquant §3.2.10 on obtient le résultat voulu. \square

§3.2.13 **Proposition.** *Soit t un terme uniforme, α une adresse active de t et τ une clique telle que $t \Gamma_\alpha \tau$. Pour tout u tel que $t \in \mathcal{T}(u)$, il existe v tel que $u \beta_\alpha^1 v$ et $\tau \subset \mathcal{T}(v)$.*

Démonstration. Comme α est une adresse active de t , alors on peut écrire $t = C[r_1]_1 \dots [r_n]_n$ avec C un contexte et pour tout i , $r_i \sqsubset_\alpha t$. Ainsi un élément de $s \in \tau$ peut s'écrire $s = C'[r'_1]_1 \dots [r'_n]_n$ où les r'_i appartiennent aux réduits des r_i . Comme t est uniforme, d'après §3.1.17 on sait qu'il existe $u \in \Lambda$ tel que $t \in \mathcal{T}(u)$. Donc α est également une adresse active de u . Donc on peut écrire u sous la forme $u = C'[r]$ et on a C est un approximant de C' et les r_i sont des approximants de r . Le résultat de la réduction de u selon α est $v = C'[r']$ où r' est le réduct de r . On doit désormais montrer que les r'_i sont des approximants de r' . On utilise pour cela les mêmes arguments que pour la proposition précédente et on obtient le résultat voulu. \square

On peut représenter cela par le diagramme suivant.

$$\begin{array}{ccc} u & \longrightarrow & t \in \mathcal{T}(u) \\ \beta_\alpha^1 \downarrow & & \downarrow \Gamma_\alpha \\ v & \longrightarrow & \tau \subset \mathcal{T}(v) \end{array}$$

Remarque : Attention on n'a pas montré que ce diagramme commutait. C'est ce qu'on va faire dans ce qui suit.

La réduction à pas de géant est bien définie pour une étape de réduction. On va maintenant la définir pour un nombre quelconque d'étapes de réduction.

On définit alors la réduction Γ_α pour une clique de termes. Soit τ une clique de termes, $t \in \tau$ et α une adresse active de t . On sait d'après ce qui vient d'être fait, qu'on peut alors réduire tous les termes de τ selon α . On notera $\Gamma_\alpha(t)$ la clique σ telle que $t \Gamma_\alpha \sigma$. Alors

$$\tau \Gamma_\alpha \bigcup_{t \in \tau} \Gamma_\alpha(t).$$

On peut donc itérer cette réduction et de ce fait la Γ -réduction en plusieurs étapes est bien définie.

§3.2.14 **Définition.** L'ensemble des termes apparaissant dans la réduction à pas de géant de t en $\text{NF}(t)$ est appelé *cône de réduction de t* . On pourra le noter $\downarrow t$.

§3.2.15 **Définition.** On appellera *cône de réduction stratifié* de t , l'ensemble de cliques de termes τ_1, \dots, τ_n tel qu'il existe des adresses $\alpha_1, \dots, \alpha_n$ telles que $t \Gamma_{\alpha_1 \dots \alpha_n} \tau_n$ et pour tout $i > 1$, $\Gamma_{\alpha_i}(\tau_{i-1}) = \tau_i$ et $\Gamma_{\alpha_1}(t) = \tau_1$.

§3.2.16 **Définition.** Soit t_0 un terme tel que $t_0 \Gamma_{\alpha_1 \dots \alpha_n} \tau$. On appellera *branche de réduction de t_0* une suite de termes t_0, t_1, \dots, t_n tels que pour tout i , $t_i \in \Gamma_{\alpha_i}(t_{i-1})$.

§3.2.17 **Proposition.** *La réduction à pas de géant est confluente et fortement normalisante.*

Démonstration. La confluence vient du fait qu'elle correspond à la β -réduction du λ -calcul pur. La normalisation forte est obtenue par le même argument que pour la β_Δ -réduction. \square

On a alors tous les outils à notre disposition pour montrer un des résultats principaux de ce travail.

§3.2.18 **Théorème.** *Soit $u \in \Lambda$, $\alpha \in \Sigma^*$ une adresse active de u et u' le réduit de u selon α . Pour tout $s' \in \mathcal{T}(u')$ il existe un unique $s \in \mathcal{T}(u)$ tel que s' appartient aux réduits à pas de géant de s selon α .*

Démonstration. Supposons donnés u, u', α et s' . L'idée est de construire s à partir de s' en utilisant la réduction de u à u' . On sait que α est active, c'est à dire que le terme à cette adresse est un redex, donc il existe r un redex et $C[\]$ un contexte tel que $r \sqsubset_\alpha u$ et $u = C[r]$. Si on note r' le réduit de r on a $u' = C[r']$ et $r' \sqsubset_\alpha u'$. Supposons que $r = (\lambda x v)w$, alors on note $X = \{\beta \in \Sigma^* \mid x \sqsubset_{\alpha c a \beta} u\}$ l'ensemble des adresses de la variable de substitution dans v . Soit $R' = [r'_i \mid r'_i \sqsubset_\alpha s']$ le multi-ensemble contenant tous les termes qui ont pour adresse α dans s' et soit n le nombre d'éléments dans R' . Pour

chaque r'_i on construit le multi-ensemble $T_{r'_i} = [t \in \Delta \mid \exists \beta \in X, t \sqsubset_\beta r'_i]$, qui est uniforme. De plus il existe $C'[\]_1 \dots [\]_n$ tel que $s' = C'[r'_{i_1}]_1 \dots [r'_{i_n}]_n$. Pour tout i on a $r'_i = C'_i[t_1]_1 \dots [t_{n_i}]_{n_i}$ avec les t_j dans $T_{r'_i}$. Soit x une variable fraîche, pour tout i on note $q_i = C'_1[x]_1 \dots [x]_{n_i}$ et $r_i = \langle \lambda x q_i \rangle T_{r'_i}$. Le terme s dont s' est un réduit est alors donné par la construction suivante :

$$s = C'[r_{i_1}]_1 \dots [r_{i_n}]_n.$$

L'unicité se déduit de §3.1.19. \square

Ce théorème montre l'existence d'un parent pour tout terme, en forme normale ou pas. Et ce, même si le terme en question ne se normalise pas. De plus par construction le terme s tel que s' appartient aux réduits à pas de géant de s selon α est unique.

Par contre l'unicité dépend de la réduction qui est faite. Autrement dit, un terme peut avoir plusieurs parents (cohérents entre eux), mais dans ce cas il est obtenu par des réductions différentes.

§3.2.19 **Exemple.** Si on note $\delta_n = \lambda x \langle x \rangle x^n$, $\Omega_n = \langle \delta_n \rangle [\delta_0.\delta_0.\delta_1 \dots \delta_{n-1}]$, alors les deux termes $t_1 = \langle \Omega_n \rangle \Omega_0$ et $t_2 = \langle \Omega_0 \rangle \Omega_n$ se réduisent en n étapes en $s = \langle \Omega_0 \rangle \Omega_0$. Or t_1 et t_2 sont cohérents. Ce qui les distingue, en tant que parents de s , c'est les adresses des redex qui ont été réduits.

Cet exemple est basé sur le fait que t_1 et t_2 sont des approximants d'un λ -terme qui n'est pas normalisable. Mais on peut aussi en trouver avec des λ -termes qui normalisent. Un exemple se situe dans la section "Musée des Horreurs" du chapitre "Relations familiales".

Par contre les éléments de la forme normale de l'approximant d'un λ -terme normalisable n'ont, eux, qu'un unique parent parmi ces approximants. Un corollaire de ce théorème est un des principaux résultats de [ER06].

§3.2.20 **Corollaire.** Soit $t, t' \in \Delta_{\mathcal{R}}^{(1)}$.

$$\text{Si } t \supset t' \text{ alors } \text{NF}(t) \supset \text{NF}(t')$$

$$\text{De plus si } t \neq t' \text{ alors } \text{NF}(t) \cap \text{NF}(t') = \emptyset$$

On vient de montrer qu'un terme en forme normale a au plus un parent dans chaque développement de Taylor d'un λ -terme. Les termes qui ne sont pas en forme normale peuvent, eux, avoir plusieurs parents. Cela vient du fait que si on réduit un terme de deux manières différentes alors on peut aboutir au même résultat, mais pour des raisons différentes. C'est pourquoi un terme qui contient des redex peut éventuellement avoir plusieurs parents. Ce qui n'est pas le cas des termes en forme normale.

§3.2.21 **Corollaire.** *Soit u un λ -terme, u' un de ses réduits (non nécessairement en forme normale) et s' un approximant de u' . Si $\text{NF}(s') \neq \emptyset$, alors il existe un unique approximant de u qui contient s' parmi ses réduits.*

Démonstration. Supposons qu'il existe deux approximants de u , s_1, s_2 qui contiennent s' parmi leurs réduits. Soit s_0 un élément de $\text{NF}(s')$. Alors s_0 est aussi un élément de $\text{NF}(s_1)$ et de $\text{NF}(s_2)$. Donc s_0 a deux parents parmi les approximants de u . Or on vient de montrer que c'est impossible. Donc s' n'a qu'un seul parent parmi les approximants de u . \square

Considérons le diagramme suivant, où la colonne de gauche représente des λ -termes purs et celle de droite leurs approximants linéaires.

$$\begin{array}{ccc} u & \longrightarrow & \mathcal{T}(u) \\ \beta^* \downarrow & & \downarrow \Gamma^* \\ u' & \longrightarrow & \mathcal{T}(u') \end{array}$$

On vient de montrer que pour u' , $\mathcal{T}(u)$ se réduit à pas de géant en $\mathcal{T}(u')$. Et pour tout $s' \in \mathcal{T}(u')$ fixé il existe un unique $s \in \mathcal{T}(u)$ tel que s' appartient aux réduits à pas de géant de s selon α . En effet c'est la réduction choisie qui détermine l'unique parent de s' dans la preuve de §3.2.18.

3.3 β -expansion

Pour définir correctement la β -expansion on se dote d'un opérateur de *dés substitution*. On notera $t]y/u[$ pour désigner l'ensemble des couples formés du terme t dans lequel un de ses sous-termes égal à u est remplacé par y , et par ce sous-terme lui-même. On le définit par induction sur t :

- $x]y/u[= \{(y, x)\}$ si $u = x$, \emptyset sinon.
- $(\lambda x s)]y/u[= \{(y, \lambda x s)\}$ si $u = \lambda x s$, $\lambda x(s]y/u[)$ sinon et si $u \in \text{Sub}_{\mathbb{L}}(s)$.
- $(\langle s \rangle T)]y/u[= \{(y, \langle s \rangle T)\}$ si $u = \langle s \rangle T$, $\langle s \rangle]y/u[T \cup \langle s \rangle (T]y/u[)$ sinon.
- $(t_1 \dots t_n)]y/u[= \bigcup_{i=1}^n t_1 \dots (t_i]y/u[\dots t_n$.

Avec les règles suivantes (où A, B sont des ensembles de couples) :

$$\begin{array}{ll}
\lambda x(s, v) \rightarrow (\lambda x s, v) & \lambda x A = \{(\lambda x a_1, a_2) \mid (a_1, a_2) \in A\} \\
\langle\langle s, v \rangle\rangle T \rightarrow \langle\langle s \rangle\rangle T, v & \langle A \rangle T = \{\langle\langle a_1 \rangle\rangle T, a_2 \mid (a_1, a_2) \in A\} \\
\langle s \rangle (T, v) \rightarrow \langle\langle s \rangle\rangle T, v & \langle s \rangle A = \{\langle\langle s \rangle\rangle a_1, a_2 \mid (a_1, a_2) \in A\} \\
s.(T, v) \rightarrow (s.T, v) & s.A = \{(s.a_1, a_2) \mid (a_1, a_2) \in A\} \\
\langle\langle s, u \rangle\rangle (T, v) \rightarrow \langle\langle s \rangle\rangle T, u.v & \langle A \rangle B = \{\langle\langle a_1 \rangle\rangle b_1, a_2.b_2 \mid (a_1, a_2) \in A, (b_1, b_2) \in B\}
\end{array}$$

§3.3.1 **Exemple.** $(\langle\langle x \rangle\rangle x \langle x \rangle x)]y/ \langle x \rangle x[= \{(\langle y \rangle \langle x \rangle x, \langle x \rangle x), (\langle\langle x \rangle\rangle x y, \langle x \rangle x)\}$.

On étend la β -expansion à un ensemble de termes C de la manière suivante : $C]x/u[= \{c]x/u[\mid c \in C\}$. On vérifie alors que la *désubstitution* réalise bien ce qu'on en attend.

§3.3.2 **Proposition.** Soit $t \in \Delta^{(1)}$ et x une de ses variables libres. Alors pour tout terme u on a $(t, u) \in (t]u/x[)]x/u[$.

Démonstration. On raisonne par induction sur t .

- Si $t = x$ alors $t]u/x[= u$. On a donc $u]x/u[= (x, u)$ qui est le résultat voulu.
- Si $t = \lambda y s$ alors $t]u/x[= \lambda y s]u/x[$. Or

$$(\lambda y s]u/x[)]x/u[= \lambda y (s]u/x[)]x/u[.$$

Par hypothèse d'induction on a $(s, u) \in (s]u/x[)]x/u[$. On en déduit le résultat.

- Si $t = \langle s \rangle T$ alors $t]u/x[= \langle s \rangle (T)]u/x[\cup \langle s]u/x[\rangle T$. Or

$$(\langle s \rangle (T)]u/x[\cup \langle s]u/x[\rangle T)]x/u[= (\langle s \rangle (T)]u/x[)]x/u[\cup (\langle s]u/x[\rangle T)]x/u[$$

et

$$\begin{array}{l}
\langle s \rangle (T]u/x[)]x/u[\subset (\langle s \rangle (T)]u/x[)]x/u[\\
\langle (s]u/x[\rangle T)]x/u[\subset (\langle s]u/x[\rangle T)]x/u[.
\end{array}$$

Par hypothèse d'induction on a

$$\begin{array}{l}
(T, u) \in (T]u/x[)]x/u[\\
(s, u) \in (s]u/x[)]x/u[
\end{array}$$

D'après les règles énoncées en début de section on a bien dans les deux cas $(\langle s \rangle T, u) \in (\langle s \rangle T]u/x[)]x/u[$.

□

§3.3.3 **Proposition.** Soit $t, u_1, u_2 \in \Delta$ et $y \in \mathcal{V} \setminus (\mathcal{V}(u_1) \cup \mathcal{V}(u_2))$, alors

$$(t]y/u_1[)]y/u_2[= (t]y/u_2[)]y/u_1[$$

On définit ensuite la "désubstitution" à grand pas. On note $t]y/U[$ le couple formé du terme t dans lequel on a remplacé chaque élément du monôme U par y , et de U . C'est à dire :

$$t]y/U[= (\dots(t]y/u_1[\dots])]y/u_n[$$

De plus $t]y/1[= (t, 1)$.

On note $\text{Exp}(u)$ les β -expansés de u tel que si u est uniforme alors $\text{Exp}(u)$ l'est aussi.

§3.3.4 **Définition.** Soit $u \in \Delta$, alors les β -expansés de u sont définis comme suit (on notera $T = [t_1 \dots t_n]$) :

- $\text{Exp}(x) = \{\langle \lambda y x \rangle 1, \langle \lambda y y \rangle x\}$
- $\text{Exp}(\lambda x u) = \{\langle \lambda y s \rangle [t_1 \dots t_n] \mid \forall i, t_i \sqsubset^L (\lambda x u), (s, T) \in (\lambda x u)]y/T[\}$
 $\cup \{\lambda x v \mid v \in \text{Exp}(u)\}$
- $\text{Exp}(\langle u \rangle V) = \{\langle \lambda y s \rangle [t_1 \dots t_n] \mid \forall i, t_i \sqsubset^L (\langle u \rangle V), (s, T) \in (\langle u \rangle V)]y/T[\}$
 $\cup \langle \text{Exp}(u) \rangle V \cup \langle u \rangle \text{Exp}(V)$
- $\text{Exp}(v_1 \dots v_n) = \{t_1 \dots t_n \mid \forall i, j, t_i \supset t_j \text{ et } t_i \in \text{Exp}(u_i)\}$

On va montrer que si deux parents d'un terme, qui ont été calculés en désubstituant deux monômes cohérents entre eux, sont cohérents alors c'est qu'ils sont égaux.

§3.3.5 **Proposition.** Soit t un terme avec ressources et T_1, T_2 deux monômes de sous-termes de t . Pour tout $(s_1, T_1) \in t]y/T_1[$, $(s_2, T_2) \in t]y/T_2[$ tels que $T_1 \supset T_2$ alors $s_1 \supset s_2$ implique $s_1 = s_2$.

Démonstration. Soit $(s_1, T_1) \in t]y/T_1[$, $(s_2, T_2) \in t]y/T_2[$ tels que $T_1 \supset T_2$. on suppose $s_1 \supset s_2$ et on fait une induction sur t .

- Si $t = x$ alors c'est immédiat ;
- Si $t = \lambda x u$ alors par définition $(s_i, T_i) \in \lambda x (u]y/T_i[)$ et il existe u_1, u_2 tels que $u_i \in u]y/T_i[$ et $s_i = \lambda x u_i$. On a donc $u_1 \supset u_2$ et par hypothèse d'induction $u_1 = u_2$. Donc $s_1 = s_2$;
- Si $t = \langle u \rangle V$ alors par définition, $t]y/T_i[= \bigcup_{T_{i1}T_{i2}=T_i} \langle u]y/T_{i1}[\rangle (V]y/T_{i2}[)$.
 Donc il existe $T_{11}, T_{12}, T_{21}, T_{22}, V_1, V_2$ des monômes et u_1, u_2 des termes

tels que

$$\left\{ \begin{array}{l} T_{12}T_{11} = T_1 \\ (u_1, T_{11}) \in u]y/T_{11}[\\ (V_1, T_{12}) \in V]y/T_{12}[\\ s_1 = \langle u_1 \rangle V_1 \end{array} \right. \quad \left\{ \begin{array}{l} T_{21}T_{22} = T_2 \\ (u_2, T_{12}) \in u]y/T_{12}[\\ (V_2, T_{22}) \in V]y/T_{22}[\\ s_2 = \langle u_2 \rangle V_2 \end{array} \right.$$

Comme $T_1 \subset T_2$, alors on a aussi $T_{11} \subset T_{21}$ et $T_{12} \subset T_{22}$. Par hypothèse d'induction on a $u_1 \subset u_2$ implique $u_1 = u_2$. De plus $(V_1, T_{12}) \in V]y/T_{12}[$ donc il existe T_{121}, \dots, T_{12n} des monômes et v_1, \dots, v_n des termes tels que :

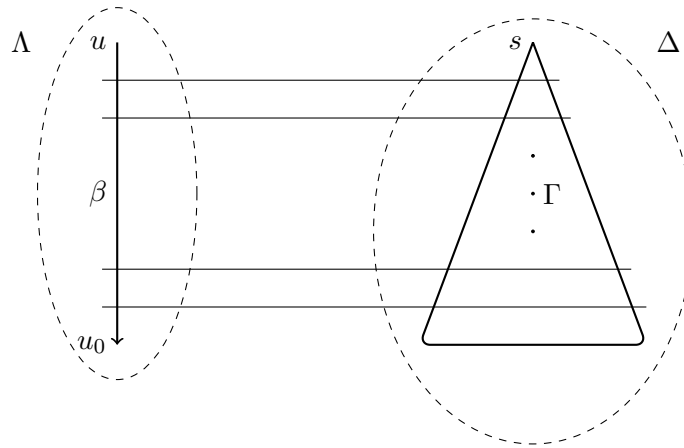
$$V]y/T_{12}[= \bigcup_{T_{121} \dots T_{12n} = T_{12}} \left(\prod_{i=1}^n v_i]y/T_{12i}[\right)$$

Donc il existe v_{11}, \dots, v_{1n} tels que $V_1 = v_{11} \dots v_{1n}$ et pour tout i , $(v_{1i}, T_{12i}) \in v_i]y/T_{12i}[$. De même pour $V_2 = v_{21} \dots v_{2n}$ avec pour tout i , $(v_{2i}, T_{22i}) \in v_i]y/T_{22i}[$. Par hypothèse d'induction on a pour tout i , $v_{1i} \subset v_{2i}$ implique $v_{1i} = v_{2i}$. On en déduit $V_1 \subset V_2$ implique $V_1 = V_2$. Donc comme $s_1 = \langle u_1 \rangle V_1$ et $s_2 = \langle u_2 \rangle V_2$, alors $s_1 \subset s_2$ implique $s_1 = s_2$. □

Chapitre 4

Relations familiales

Le but de ce chapitre est d'étudier les liens qui existent entre les différents termes qui appartiennent au même cône de réduction et plus particulièrement aux termes cohérents qui sont issus d'une même réduction. On a vu précédemment qu'il y avait un lien entre la Γ -réduction et la β -réduction. Si on prend un terme avec ressources t qui est uniforme, on peut lui associer un cône de réduction stratifié correspondant à la β -réduction. Voyons cela sur un dessin.



Une ligne horizontale sur le dessin représente un ensemble d'approximants du λ -terme pur correspondant. Entre deux lignes il y a une étape de réduction (Γ à droite et β à gauche).

Dans ce chapitre on utilisera le terme réduction pour désigner la réduction β_Δ . On précisera réduction à pas de géant si on veut parler de Γ . Tous les termes de ce chapitre sont uniformes.

4.1 Frères

On s'intéresse tout d'abord aux termes qui sont obtenus en une seule étape de réduction (la généralisation pour étape de Γ -réduction est immédiate). On voudrait répondre à la question suivante : Si on se donne deux termes t_1, t_2 (cohérents) peut on savoir si ils appartiennent aux réduits, en une étape de réduction, d'un seul et même terme ? Commençons par définir correctement la notion qui nous intéresse.

§4.1.1 **Définition.** Deux termes t_1, t_2 sont dits *frères* si il existe une variable x , un terme simple s , un monôme T et un contexte $C[\]$ tels que $t_1, t_2 \in \{C[t] \mid t \in \partial_x(s, T)\}$.

§4.1.2 **Théorème.** Soit u, v deux termes frères, alors il existe un entier n , un contexte linéaire $C[\]_1 \dots [\]_n$, t_1, \dots, t_n des sous-termes de u deux à deux cohérents et une permutation σ de $1, \dots, n$ tels que $u = C[t_1]_1 \dots [t_n]_n$ et $v = C[t_{\sigma(1)}]_1 \dots [t_{\sigma(n)}]_n$.

Démonstration. La réduction passe au contexte donc on peut se restreindre à la réduction d'un redex. Soit $t = \langle \lambda x s \rangle [t_1 \dots t_n]$. Les réduits de ce redex appartiennent tous, par définition, à $s[t_1/x] \dots [t_n/x]$. Donc ils diffèrent seulement par l'emplacement des t_1, \dots, t_n . \square

Ce théorème donne une caractérisation des termes frères.

Donc la fratrie de deux termes est décidable et de plus la preuve est constructive. C'est à dire que si on a prouvé que deux termes sont frères alors on peut déduire de la preuve un parent commun. Par contre il n'y a pas unicité du parent commun.

Regardons la notion de frères comme une relation.

§4.1.3 **Proposition.** La relation de frère est réflexive et symétrique mais pas transitive.

Démonstration. Elle est trivialement réflexive et symétrique. La non transitivité vient de la non transitivité de la cohérence. En effet prenons $t_1 = \langle \langle x \rangle 1 \rangle \langle \langle x \rangle x \rangle \langle x \rangle y$, $t_2 = \langle \langle x \rangle x \rangle \langle \langle x \rangle 1 \rangle \langle x \rangle y$, $t_3 = \langle \langle x \rangle y \rangle \langle \langle x \rangle x \rangle \langle x \rangle 1$. On a t_1 qui est frère avec t_2 et avec t_3 . Par contre t_2 et t_3 ne peuvent pas être frères car ils ne sont pas cohérents (on sait d'après le chapitre précédent que les réduits d'un même terme sont cohérents entre eux). \square

Même si on restreint cette relation à des termes cohérents, on n'a pas la transitivité.

- §4.1.4 **Exemple.** (i). $\langle\langle\langle\star\rangle\langle x\rangle x^2\rangle y^2\rangle\langle\langle\star\rangle\langle x\rangle x\rangle y$
(ii). $\langle\langle\langle\star\rangle\langle x\rangle x\rangle y^2\rangle\langle\langle\star\rangle\langle x\rangle x^2\rangle y$
(iii). $\langle\langle\langle\star\rangle\langle x\rangle x^2\rangle y\rangle\langle\langle\star\rangle\langle x\rangle x\rangle y^2$

Ces trois termes sont cohérents, (ii) est frère avec (i) : un parent commun est $\langle\lambda z\langle\langle\langle\star\rangle z\rangle y^2\rangle\langle\langle\star\rangle z\rangle y\rangle[\langle x\rangle x.\langle x\rangle x^2]$, et (ii) est frère avec (iii) : un parent commun est $\langle\lambda z\langle z\rangle z\rangle[\langle\langle\star\rangle\langle x\rangle x\rangle y^2.\langle\langle\star\rangle\langle x\rangle x^2\rangle y]$, mais (i) n'est pas frère avec (iii). On peut le vérifier en calculant $\text{Exp}(i) \cap \text{Exp}(iii)$.

On peut naturellement étendre cette relation binaire à une relation n -aire.

4.2 Cousins

On a une caractérisation complète des termes frères, on peut maintenant étendre cette définition en autorisant plusieurs étape de réduction pour le parent commun. Définissons cette nouvelle notion.

- §4.2.1 **Définition.** Les termes u, v sont dits *cousins* si il existe un terme s et une clique de termes τ appartenant à un cône de réduction de s , tels que $u, v \in \tau$.

Remarque : Si u et v sont cousins alors $u \subset v$.

Cette définition englobe la précédente mais elle est plus générale. On peut montrer que deux termes peuvent être cousins alors qu'on ne peut pas obtenir l'un des deux en permutant certains sous-termes de l'autre, ce qui est le cas si les termes sont frères. Pour des exemples de termes cousins on pourra se reporter à la section 4.7.

Montrons quelques propriétés et contre-propriétés des termes cousins.

- §4.2.2 **Proposition.** *Deux termes cousins ont le même nombre d'occurrences de chaque variable libre.*

Démonstration. Le nombre d'occurrences des variables libres est invariant par réduction. \square

- §4.2.3 **Contre-Proposition.** *Soit t un terme avec ressources. Les branches de réductions de t ne sont pas toutes de la même taille.*

Démonstration. Bien que la réduction soit linéaire et qu'on ne puisse ni dupliquer les ressources, ni les supprimer, on peut toutefois durant la réduction créer un redex dans une des branches de réduction et pas dans une autre. Un exemple se trouve en §4.7.3 \square

- §4.2.4 **Corollaire.** *Deux termes cousins n'ont pas nécessairement la même taille.*

La relation de cousins passe au contexte.

§4.2.5 **Proposition.** Soit t_1, t_2 deux termes cousins. Alors pour tout contexte $C[]$, $C[t_1]$ et $C[t_2]$ sont également cousins.

Démonstration. Soit t un parent de t_1 et t_2 . Comme la réduction passe au contexte alors pour tout $s \in \downarrow t$ et pour tout contexte $C[]$ on a $C[s] \in \downarrow C[t]$. En particulier pour t_1 et t_2 . \square

En étudiant la réduction, on remarque qu'à chaque étape un monôme de taille n ainsi qu'une abstraction de degré n "disparaissent". Afin d'avoir un résultat qui rende compte de ce phénomène, on doit introduire de nouveaux objets.

§4.2.6 **Définition.** Soit s un terme. On définit inductivement le multi-ensemble d'entiers $\gamma(s)$ noté multiplicativement, c'est à dire de la forme $[0^{k_0} \dots n^{k_n}]$, qui représente le nombre d'applications dans s ainsi que la taille de leurs arguments.

- $\gamma(x) = 0$
- $\gamma(\lambda x u) = \gamma(u)$
- $\gamma(\langle s \rangle t_1 \dots t_n) = n^1 \cdot \gamma(s) \cdot \prod_{i=1}^n \gamma(t_i)$

La somme des exposants donne le nombre d'applications dans s et k_i représente le nombre de monômes de taille i .

§4.2.7 **Exemple.** Soit $t = \lambda x \langle \lambda y \langle \langle x \rangle y \rangle 1 \rangle \langle x \rangle x^5$. On a $\gamma(t) = [0^1 \cdot 1^2 \cdot 5^1]$.

§4.2.8 **Définition.** Soit s un terme. On définit inductivement le multi-ensemble d'entiers $\delta(s)$ noté multiplicativement, c'est à dire de la forme $[0^{k_0} \dots n^{k_n}]$, qui représente le nombre de sous-termes de la forme $\lambda x u$ dans s ainsi que le degré de la variable liée.

- $\delta(x) = 0$
- $\delta(\lambda x s) = \text{deg}_x(s)^1 \cdot \delta(s)$
- $\delta(\langle s \rangle t_1 \dots t_n) = \delta(s) \cdot \prod_{i=1}^n \delta(t_i)$

La somme des exposants donne le nombre d'abstractions dans le terme et k_i correspond au nombre d'abstractions qui lient i occurrences de variables.

§4.2.9 **Exemple.** Soit $t = \lambda x \langle \lambda y \langle \langle x \rangle y \rangle 1 \rangle \langle x \rangle x^5$. On a $\delta(t) = [1^1 \cdot 7^1]$.

§4.2.10 **Définition.** Soit s un terme. On définit le multi-ensemble d'entiers $\xi(s)$ comme le quotient formel de $\gamma(s)$ par $\delta(s)$. Soit k_0, \dots, k_n les exposants de $\gamma(s)$ et k'_0, \dots, k'_l les exposants de $\delta(s)$ alors $\xi(s)$ est de la forme $[0^{k_0 - k'_0} \dots m^{k_m - k'_m}]$ avec m le max de n et l . Par convention $k_j = 0$ (respectivement $k'_j = 0$) si j n'appartient pas à $\gamma(s)$ (respectivement $\delta(s)$). On notera $\xi(s) = \frac{\gamma(s)}{\delta(s)}$.

§4.2.11 **Exemple.** Soit $t = \lambda x \langle \lambda y \langle \langle x \rangle y \rangle 1 \rangle \langle x \rangle x^5$. On a

$$\xi(t) = \frac{\gamma(t)}{\delta(t)} = [0^{1-0} \cdot 1^{2-1} \cdot 5^{1-0} \cdot 7^{0-1}] = [0^1 \cdot 1^1 \cdot 5^1 \cdot 7^{-1}].$$

On va montrer que ξ est un invariant de réduction

§4.2.12 **Théorème.** Soit s un terme avec ressources et T un monôme. Le multi-ensemble $\xi(\langle \lambda x s \rangle T)$ est un invariant pour la réduction.

Démonstration. Montrons que pour t dans $\partial_x(s, T)$, $\xi(t) = \xi(\langle \lambda x s \rangle T)$. Commençons par calculer chaque multi-ensemble en supposant que $T = [t_1 \dots t_n]$.

$$\begin{aligned} \gamma(\langle \lambda x s \rangle T) &= |T|^1 \cdot \gamma(s) \cdot \prod_{i=0}^n \gamma(t_i) \\ \delta(\langle \lambda x s \rangle T) &= \text{deg}_x(s)^1 \cdot \delta(s) \cdot \prod_{i=0}^n \delta(t_i) \\ \xi(\langle \lambda x s \rangle T) &= \frac{\gamma(\langle \lambda x s \rangle T)}{\delta(\langle \lambda x s \rangle T)} \\ &= \frac{\gamma(s)}{\delta(s)} \cdot \prod_{i=0}^n \frac{\gamma(t_i)}{\delta(t_i)} \cdot \frac{|T|}{\text{deg}_x(s)} \\ &= \xi(s) \cdot \prod_{i=0}^n \xi(t_i) \cdot \frac{|T|}{\text{deg}_x(s)} \end{aligned}$$

Or $\xi(t) = \xi(s) \cdot \prod_{i=0}^n \xi(t_i)$ et $|T| = \text{deg}_x(s)$ (sinon $t = \emptyset$). Donc on obtient bien l'égalité voulue : $\xi(t) = \xi(\langle \lambda x s \rangle T)$. On remarque que réduire un redex dans un terme revient à enlever un élément dans γ et dans δ , mais surtout que cet élément est le même, si l'ensemble des réduits est non vide. \square

§4.2.13 **Corollaire.** Pour tout t dans le cône de réduction de s on a $\xi(t) = \xi(s)$.

Pour autant ξ ne caractérise pas un cône de réduction. Un exemple est donné dans la suite.

§4.2.14 **Corollaire.** Si t_1 et t_2 sont cousins alors $\xi(t_1) = \xi(t_2)$.

Par contre la réciproque n'est pas vraie.

§4.2.15 **Exemple.** Soit $u = \langle \langle x \rangle y^2 \rangle x$ et $v = \langle \langle x \rangle y \rangle x^2$. Ces deux termes ont le même ξ mais pas le même nombre d'occurrences de leurs variables libres. Donc ils ne sont pas cousins.

Soit t un terme, $\gamma(t)$, $\delta(t)$ et $\xi(t)$ permettent d'avoir des informations sur les éléments de la forme normale de t et le nombre d'étapes de réduction pour y arriver. Les exposants positifs (respectivement négatifs) de $\xi(t)$ donnent le nombre minimum d'applications (respectivement d'abstractions), et la taille de leurs arguments (respectivement le nombre d'occurrences de variables qu'elles lient), que doivent contenir les éléments de la forme normale de t . De plus en prenant le minimum de la différence entre la somme des exposants de $\gamma(t)$ et la somme des exposants positifs de $\xi(t)$, et la différence entre la somme des exposants de $\delta(t)$ et la somme des exposants négatifs de $\xi(t)$ on obtient le nombre d'étapes maximum de réduction pour atteindre la forme normale. Si on reprend l'exemple §4.2.11, la différence entre la somme des exposants de $\gamma(t)$ et la somme des exposants positifs de $\xi(t)$ est de 2 et la différence entre la somme des exposants de $\delta(t)$ et la somme des exposants négatifs de $\xi(t)$ est de 1. Donc il faut faire au plus une réduction pour obtenir la forme normale.

On peut également utiliser ξ pour déterminer une partie des termes qui se réduisent en \emptyset . Dans les cas où la forme normale du λ -terme approximé est assez simple, l'ensemble des ξ des approximants est inclus strictement dans l'ensemble des ξ possibles. Donc si t est un approximant de u et que la forme normale de u est assez simple (le cas le plus évident est $*$) alors si $\xi(t)$ ne fait pas partie des ξ possibles pour les approximants de la forme normale (ici le seul ξ possible est $[\]$) c'est qu'il se réduit en \emptyset . L'avantage c'est que c'est beaucoup plus rapide de calculer $\xi(t)$ que de réduire t .

4.3 Les termes sans λ et sans monôme vide

Dans cette section on étudiera le cas des termes qui ne contiennent pas de λ , ni de monôme vide. Remarquons que pour un tel terme t on a $\xi(t) = \gamma(t)$. Dans toute cette section on désignera par «terme» un λ -terme avec ressources ne contenant ni λ , ni monôme vide.

§4.3.1 **Proposition.** *Soit t un terme. Si t' est cohérent avec t alors t' ne contient pas non plus de λ .*

Démonstration. On procède par induction sur t .

- Si $t = x$ le résultat est immédiat.
- Si $t = \langle s \rangle T$ alors t' est de la forme $\langle s' \rangle T'$ avec $s \subset s'$ et $T \subset T'$. Comme t ne contient pas de λ ni de monôme vide, c'est aussi le cas de s et de T , en particulier T n'est pas vide. Soit $T = t_1 \dots t_n$. Par

hypothèse d'induction s' et T' ne contiennent pas de λ ni de monôme vide. Donc c'est aussi le cas de $\langle s' \rangle T' = t'$. \square

§4.3.2 **Corollaire.** *Si t ne contient pas de λ , ni de monôme vide alors c'est aussi le cas de ses cousins.*

§4.3.3 **Proposition.** *Soit t un terme alors ses cousins ont la même taille que t .*

Démonstration. On sait que deux termes cousins doivent avoir le même ξ . Par définition $\xi(t) = \frac{\gamma(t)}{\delta(t)}$, or comme t ne contient pas de λ on a $\delta(t) = []$. Soit t' un cousin de t , on a $\xi(t) = \xi(t')$. Mais $\xi(t) = \gamma(t)$ et $\xi(t') = \gamma(t')$ puisque d'après §4.3.1 t' n'a pas non plus de λ , donc on a $\gamma(t) = \gamma(t')$. On a montré précédemment que deux termes cousins avaient le même nombre de variables libres, or dans ce cas toutes les variables sont des variables libres. De plus on peut déduire de $\gamma(t)$ le nombre d'applications que contient t en faisant la somme de tous les exposants. Donc t et t' ont le même nombre d'applications et le même nombre de variables libres et comme ils ne contiennent pas de λ alors ils ont la même taille. \square

§4.3.4 **Définition.** Notons $\mathcal{F}(t)$ l'ensemble des termes qui sont cousins avec t .

On remarquera que cet ensemble n'est pas une clique en général.

§4.3.5 **Lemme.** *Soit t un terme alors l'ensemble des termes cousins avec t , $\mathcal{F}(t)$, est fini.*

Démonstration. L'ensemble des termes cousins avec t est inclus dans l'ensemble des termes cohérents avec t et de même taille que t . Cet ensemble est fini car tous ses éléments sont constitués des mêmes variables. \square

Remarque : On ne sait pas si cet ensemble est une clique pour la relation de cousins.

§4.3.6 **Contre-Proposition.** *Soit t_1 et t_2 deux termes. Si t_1 est cohérent avec t_2 et que $\gamma(t_1) = \gamma(t_2)$ alors cela n'implique pas que t_1 et t_2 soient cousins.*

§4.3.7 **Exemple.** Prenons simplement $t_1 = \langle \langle x \rangle x^2 \rangle \langle y \rangle y$ et $t_2 = \langle \langle x \rangle x \rangle \langle y \rangle y^2$. On voit bien que ces deux termes sont cohérents et qu'ils ont le même γ mais ils n'ont pas les mêmes variables libres, t_1 a trois occurrences de x alors que t_2 n'en a que deux.

Cet exemple montre également qu'un même ξ ne caractérise pas qu'un seul cône de réduction. Tous les termes à l'intérieur d'un cône de réduction ont le même ξ , mais plusieurs cônes de réduction distincts (qui ne sont pas inclus dans un cône plus grand) peuvent avoir le même ξ .

4.4 Les termes sans monômes vides

Après avoir étudié les termes qui ne contenaient que des applications et sans monôme vide, on peut généraliser un peu et s'intéresser aux termes dont la seule contrainte est de ne pas contenir de monôme vide.

§4.4.1 **Proposition.** *Soit t un λ -terme avec ressources dont tous les monômes sont non vides, alors il existe un unique λ -terme pur u tel que t est un approximant de u .*

Démonstration. Montrons ce résultat par induction sur t .

- Si $t = x$ alors le résultat est immédiat.
- Si $t = \lambda x s$ alors par hypothèse les monômes de s sont tous non vides et par hypothèse d'induction il existe un unique λ -terme v tel que $u \in \mathcal{T}(v)$. Soit le λ -terme $u = \lambda x v$, alors par construction t est un approximant de u et par unicité de v , u est l'unique λ -terme tel que $t \in \mathcal{T}(u)$.
- Si $t = \langle s \rangle T$ alors puisque t ne contient pas de monôme vide, c'est aussi le cas de s et de tous les éléments de T . Par hypothèse d'induction il existe un unique λ -terme v tel que $s \in \mathcal{T}(v)$ et pour tout élément $s_i \in T$ il existe un unique λ -terme w_i dont t_i est un approximant. Or d'après §3.1.17 on sait qu'il existe un λ -terme w tel que $T \subset \mathcal{T}(w)$. Donc tous les w_i sont identiques et égaux à w , qui est donc unique. On construit le terme $u = (v)w$ dont t est un approximant par définition et qui est unique par construction.

De plus cette preuve est constructive. □

§4.4.2 **Corollaire.** *La relation de cohérence restreinte aux termes sans monômes vides est transitive.*

Le simple fait de rajouter les λ autorise les termes cousins à ne pas avoir la même taille. Un exemple si situé en §4.7.4.

4.5 Les termes dont tous les monômes ont la même taille

Considérons les termes dont tous les monômes sont de taille n . La syntaxe de ces termes est la suivante :

$$s, t_1, \dots, t_n := x \mid \lambda x s \mid \langle s \rangle [t_1 \dots t_n]$$

§4.5.1 **Proposition.** *Soit t, u deux termes n'ayant que des monômes de taille n . Alors $u \subset t$ implique $u = t$.*

Démonstration. Soit t, u deux termes n'ayant que des monômes de taille n . On suppose $u \subset t$ et on montre par induction sur t que $t = u$.

- Si $t = x$ alors le résultat est immédiat ;
- Si $t = \lambda x s$ alors par cohérence $u = \lambda x v$ et par hypothèse d'induction on a $s = v$ ce qui nous donne l'égalité voulue ;
- Si $t = \langle s \rangle [t_1 \dots t_n]$ alors par cohérence on a $u = \langle v \rangle [u_1 \dots u_n]$, de plus $s \subset v$ donc par hypothèse d'induction on a $s = v$. De même pour tout $j \in \{1, \dots, n\}$, $u_i \subset t_j$, donc par hypothèse d'induction on a pour tout i, j , $u_i = t_j$. Ce qui donne $u = t$.

□

La syntaxe des termes dont tous les monômes sont de taille n énoncée plus haut se simplifie alors en :

$$s, t := x \mid \lambda x s \mid \langle s \rangle t^n$$

On peut déduire de cette proposition le résultat suivant.

§4.5.2 **Proposition.** *Un terme qui ne contient pas de λ et qui n'a que des monômes de taille $n > 0$ n'a pas de cousins.*

Démonstration. Soit t un tel terme. Supposons que t' soit un cousin de t . On a montré que dans ce cas t' ne contenait pas non plus de λ . De plus on doit avoir $\gamma(t) = \gamma(t')$, or il existe deux entiers n, k tels que $\gamma(t) = [n^k]$. D'après la proposition précédente on a alors $t = t'$. □

Remarque : C'est le cas de l'approximant canonique (dont tous les monômes ont une taille 1) lorsqu'il ne contient pas d'abstraction.

4.6 Décidabilité du cousinage

On a vu précédemment que la relation de frères était décidable. Pour les termes cousins c'est un peu plus compliqué.

Si on se fixe deux termes avec ressources on voudrait savoir si leur cousinage est décidable. Autrement dit si on peut construire un algorithme qui prend comme arguments ces deux termes et qui répond «oui» si ils sont cousins et «non» si il ne le sont pas.

Pour l'instant on a seulement la semi décidabilité. En effet le nombre de β -expansés en une étape étant fini, si un parent commun existe en n étapes de réduction alors il se trouve parmi un nombre fini de termes. En explorant tous les β -expansés des deux termes de départ on le trouve en temps fini. Mais on ne sait pas, dans le cas où un parent commun existerait, combien d'étapes de réduction il lui faut pour obtenir nos deux termes de départ. Donc on ne peut pas faire mieux que d'explorer tous les β -expansés. Comme il y en a un nombre infini, on sait que la relation de cousins est semi-décidable. En effet si il existe un parent commun alors on va le trouver, mais si un tel parent n'existe pas alors on continuera de chercher infiniment.

4.7 Musée des horreurs

Au cours de ce travail on a croisé quelques «monstres». Ils infirment certaines théories ou sont juste intéressants de par leur construction.

- 1 **Horreur.** *Deux termes cousins n'ont pas nécessairement les mêmes sous-termes cohérents.*

§4.7.1 **Exemple.** Les deux termes suivants sont cousins :

- (i). $\langle \langle \star \rangle \lambda x \langle \langle \star \rangle x^2 \rangle x^2 \rangle (\lambda x \langle \langle \star \rangle x^2 \rangle x^2)^3$
- (ii). $\langle \langle \star \rangle [\lambda x \langle \langle \star \rangle x^2 \rangle x^2 . \lambda x \langle \langle \star \rangle x \rangle x^3] \rangle (\lambda x \langle \langle \star \rangle x^2 \rangle x^2)^2$

Un parent de ces deux cousins est :

$$\langle \lambda y \langle y \rangle y^4 \rangle \left[(\lambda x \langle \langle \star \rangle x^2 \rangle x^2)^4 . \lambda x \langle \langle \star \rangle x \rangle x^3 \right]$$

Quand le premier a tous ses sous-termes commençant par une abstraction égaux à $\lambda x \langle \langle \star \rangle x^2 \rangle x^2$, le deuxième en contient un qui est différent : $\lambda x \langle \langle \star \rangle x \rangle x^3$.

- 2 **Horreur.** *Deux termes cousins n'ont pas forcément la même taille.*

§4.7.2 **Exemple.** Les deux termes suivants n'ont pas la même taille et pourtant sont cousins.

- (i). $\langle\langle\langle\star\rangle\langle\star\rangle 1\rangle 1\rangle \lambda x \langle x \rangle 1\rangle \lambda y \langle\langle\langle y \rangle \lambda z \langle\star\rangle 1\rangle \lambda z \langle\star\rangle z\rangle y$
- (ii). $\langle\langle\langle\star\rangle 1\rangle \lambda z \langle\star\rangle 1\rangle \lambda x \langle x \rangle \langle x \rangle 1\rangle \lambda y \langle\langle\langle y \rangle [\lambda z \langle\star\rangle 1 . \lambda z \langle\star\rangle z]\rangle 1\rangle y$

Un des parents de ces termes est :

$$\langle\langle\lambda u \langle u \rangle [a_1 . a_2]\rangle u\rangle [\lambda y \langle\langle\langle y \rangle [b_1 . b_2]\rangle 1\rangle y . \lambda y \langle\langle\langle y \rangle b_1 \rangle b_2\rangle y]$$

avec

$$a_1 = \lambda a \langle a \rangle \langle a \rangle 1$$

$$a_2 = \lambda a \langle a \rangle 1$$

$$b_1 = \lambda b \langle\star\rangle 1$$

$$b_2 = \lambda b \langle\star\rangle b$$

3 **Horreur.** *Un terme sans abstractions peut avoir des cousins qui en contiennent.*

§4.7.3 **Exemple.** Les deux termes suivants sont cousins :

- (i). $\langle\langle\star\rangle 1\rangle \langle\langle\star\rangle \star\rangle 1$
- (ii). $\langle\langle\star\rangle \lambda x \langle\langle\star\rangle 1\rangle \langle x \rangle \star\rangle 1$

Un parent de ces deux termes est :

$$\langle\lambda y \langle y \rangle y\rangle [\lambda x \langle\langle\star\rangle 1\rangle \langle x \rangle \star . \lambda x \langle\langle\star\rangle x\rangle 1]$$

Remarque : Ce dernier exemple sert aussi d'exemple plus simple pour l'horreur précédente.

4 **Horreur.** *Deux termes cousins peuvent ne pas avoir la même taille bien qu'ayant été obtenus par le même nombre de réductions.*

§4.7.4 **Exemple.** Les deux termes suivants sont cousins.

- (i). $\langle\langle\langle\star\rangle \star^2\rangle \star\rangle \lambda y \langle\star\rangle y$
- (ii). $\langle\langle\langle\star\rangle \star\rangle \star^2\rangle \lambda y \langle\star\rangle y^2$

Un parent commun à ces deux termes est :

$$\langle\lambda z \langle\langle\lambda u \langle\lambda x \langle x \rangle \langle x \rangle z\rangle [\lambda v \langle v \rangle u . \lambda v \langle v \rangle u^2]\rangle \star^3\rangle z\rangle [\lambda y \langle\star\rangle y . \lambda y \langle\star\rangle y^2]$$

Chaque branche de réduction allant de ce parent commun à (i) ou (ii) est de taille 6. Or on a $\text{taille}(i) = 11$ et $\text{taille}(ii) = 12$. Les deux exemples précédents étaient basés sur des branches de réduction de taille différentes.

5 **Horreur.** *Un terme qui se réduit en \emptyset peut avoir plusieurs parents approxi-
mants du même terme.*

§4.7.5 **Exemple.** Le λ -terme $u = (\lambda f(f)\lambda x(f)\lambda d x)(\lambda r(r)(r)*$ se réduit en plu-
sieurs étapes en $u_1 = (\lambda d*)(\lambda d*)*$. L'approximant canonique de u_1 est
 $s_1 = \langle \lambda d* \rangle \langle \lambda d* \rangle *$. Les deux λ -termes avec ressources suivants sont des
approximants de u et tous les deux ont une branche de réduction qui mène
à s_1 .

$$(i). \langle \lambda f \langle f \rangle [t_1 . (t_2)^2] \rangle [(\lambda r \langle r \rangle \langle r \rangle 1)^2 . \lambda r \langle r \rangle 1 . \lambda r \langle r \rangle [\langle r \rangle * . \langle r \rangle *^2]]$$

$$(ii). \langle \lambda f \langle f \rangle [t_1 . t_2] \rangle [\lambda r \langle r \rangle 1 . \lambda r \langle r \rangle \langle r \rangle * . \lambda r \langle r \rangle \langle r \rangle *^2]$$

Avec $t_1 = \lambda x \langle f \rangle \lambda d x$ et $t_2 = \lambda x \langle f \rangle (\lambda d x)^2$.

6 **Horreur.** *Soit un contexte $C[\]$ et deux termes t_1, t_2 alors $C[t_1]$ et $C[t_2]$
sont cousins n'implique pas que t_1 et t_2 le soient.*

§4.7.6 **Exemple.** Le terme

$$\langle \lambda z \langle z \rangle \langle \star \rangle z \rangle [\lambda x \lambda y \langle x \rangle y . \lambda x \lambda y \langle x \rangle y^2]$$

se réduit (après α -conversion) en les deux termes suivants :

$$(i). \lambda a \langle \langle \star \rangle \lambda x \lambda y \langle x \rangle y \rangle a^2$$

$$(ii). \lambda a \langle \langle \star \rangle \lambda x \lambda y \langle x \rangle y^2 \rangle a$$

Ces deux termes sont donc cousins et de la forme $C[t_1], C[t_2]$ avec $C[\] =$
 $\lambda a[\]$. Par contre t_1 et t_2 ne sont pas cousins car ils n'ont pas le même
nombre d'occurrences de la variable libre a .

Quelques exemples en vrac

§4.7.7 **Exemple.** Soit $T = t_1, \dots, t_n$ un monôme. Les termes suivant sont cousins

$$(i). \langle \langle \star \rangle [t_1 \dots t_k] \rangle [t_{k+1} \dots t_n]$$

$$(ii). \langle \langle \star \rangle [t_{\sigma(k+1)} \dots t_{\sigma(n)}] \rangle [t_{\sigma(1)} \dots t_{\sigma(k)}]$$

de parent :

$$\langle \lambda u \langle \lambda x \langle x \rangle \langle x \rangle \star \rangle [\lambda y \langle y \rangle u^{n-k} . \lambda y \langle y \rangle u^k] \rangle T$$

pour tout n, k deux entiers possiblement nuls.

§4.7.8 **Exemple.** Les termes suivant sont cousins

$$(i). \langle \langle u \rangle v^n \rangle \langle w \rangle v^k$$

$$(ii). \langle \langle u \rangle v^k \rangle \langle w \rangle v^n$$

de parent :

$$\langle \lambda x \langle \langle x \rangle u \rangle \langle x \rangle w \rangle [\lambda y \langle y \rangle v^k . \lambda y \langle y \rangle v^n]$$

Chapitre 5

Programmation

Afin de mieux comprendre la réduction dans les termes avec ressources on a écrit un programme qui permet de réduire et d'expanser les termes du λ -calcul avec ressources. Puis on a implémenté l'extension du résultat de Ehrhard-Regnier (§3.2.18) dans ce cadre. On a donc un programme qui nous permet de réaliser effectivement ce théorème. Pour faire cela on a repris le programme qui avait été réalisé par Laurent Regnier pour démontrer un des résultats principaux de [ER06] et qui est disponible à l'adresse suivante : `iml.univ-mrs.fr/~regnier/taylor`.

Ce dernier permet de faire deux choses. D'une part il permet de normaliser un λ -terme pur. D'autre part, si on lui donne comme argument un λ -terme u et un approximant de sa forme normale t alors il permet de retrouver l'unique approximant de u qui contient t parmi ses réduits.

Nous avons repris une grande partie de ce programme pour écrire le nôtre. On n'a toutefois repris que les fonctions de base et pas la fonction de réduction d'un λ -terme jusqu'à la forme normale qui utilisait la machine de Krivine. Ce choix est motivé par le fait que l'on voulait pouvoir réduire un terme comme on le souhaitait, en particulier pas nécessairement jusqu'à la forme normale et on voulait également pouvoir choisir le redex réduit à chaque étape. La réduction littérale est l'option qui nous a paru la plus simple compte tenu de ce que l'on voulait faire.

De plus on a ajouté la possibilité pour un terme de contenir des variables libres afin d'être le plus général possible.

Les variables sont codées avec les indices de De Bruijn. Ce qui demande une gestion particulière, notamment lors de la substitution. Lorsque l'on veut réduire le terme $\langle \lambda x s \rangle t$, où s ne contient qu'une seule occurrence de x et où t est un terme, on doit compter le nombre de lambdas qui sont devant x dans

s et incrémenter d'autant toutes les variables liées de t . Évidemment ce cas est le plus simple, lorsque t est un monôme c'est un peu plus compliqué. Le même problème se pose lorsqu'on veut faire de la β -expansion, mais dans l'autre sens. On code les variables libres en leur donnant un indice de De Bruijn négatif (-2).

Comme la définition des λ -termes avec ressources est en deux parties, une pour les termes et une pour les monômes, à chaque fois qu'on écrit une fonction qui repose sur le principe d'induction sur les termes, on est obligé d'écrire également une fonction qui le fait pour les monômes.

Pour que les résultats soient plus lisibles on a écrit plusieurs fonctions d'affichage sur les termes. La première fait de l'alpha conversion afin d'éviter, lors de la réduction, que deux abstractions liant la même variable se retrouvent imbriquées. Pour cela on lit le terme et on note les noms des variables liées au moment de la lecture. Si on rencontre deux fois la même variable alors on indice la deuxième. Par exemple, $\lambda x \lambda x(x)x$ peut devenir $\lambda x \lambda x_1(x)x_1$. La deuxième fonction permet, lors de la β -expansion (Cf 5.4), de réécrire les expansés comme des approximants des λ -termes purs dont on dispose. Pour faire ceci on suit juste la définition des approximants.

5.1 Forme normale

Pour commencer on a créé un programme qui permet de calculer la forme normale d'un λ -terme avec ressources.

Pour faire cela on suit la définition de la réduction. On commence par construire une fonction qui substitue une occurrence de la variable x par un terme t de toutes les façons possibles. On déconstruit le terme de départ jusqu'à arriver sur x , puis on le reconstruit en remplaçant x par t . Pendant la phase de déconstruction, si on arrive sur une application de la forme $\langle u \rangle v$, alors on teste la présence de x dans u puis dans v . Si x apparaît dans u alors on utilise la même fonction pour substituer x dans u et on applique chaque élément du résultat à v . On fait pareil pour v . Le résultat final est la concaténation de ces deux résultats.

On utilise cette fonction (*subst_ss*) pour en faire une autre (*subst*) qui substitue une variable donnée par un monôme. Concrètement, si on veut réduire un terme de la forme $C[(\lambda x s) T]$ alors on doit décrémenter de 1 l'indice de De Bruijn de toutes les variables qui sont liées dans le contexte puisque le λx va disparaître et seulement ensuite substituer les occurrences de x par les termes présents dans le monôme T . Si on fait ça alors certaines variables de s qui sont liées dans le contexte vont se retrouver, d'après leur indice de De

Bruijn, comme des variables de substitution. Pour palier à ce problème on commence par changer la valeur de l'indice de De Bruijn des occurrences de la variable de substitution en -1 et ensuite on décrémente l'indice des variables concernées. Il ne reste plus qu'à substituer les variables dont l'indice de De Bruijn vaut -1 .

On utilise ces deux dernières fonctions pour écrire la fonction *normal_form* qui prend un terme avec ressources et qui renvoie sa forme normale. La fonction *normal_form* prend comme argument un terme avec ressources de la forme $C[\langle \lambda x s \rangle T]$ et elle appelle *subst* en lui donnant comme argument le terme s dans lequel les occurrences de la variable x ont été "marquées" et le monôme T .

Ainsi on a une fonction qui est capable de calculer la forme normale d'un λ -terme avec ressources.

5.2 β -expansion

On aimerait avoir un programme qui détermine si deux termes sont cousins. Autrement dit qui exhibe un parent commun. Pour faire cela on calcule tous les parents d'un terme en un nombre d'étapes de réduction fixé.

Afin de mieux comprendre les cousins, on a réalisé un programme capable de construire les antécédents d'un terme donné. Le but étant bien sûr de se servir de ce programme pour tester si deux termes sont cousins ou pas.

Pour faire de la β -expansion sur les termes avec ressources, il faut comprendre comment fonctionne la β -réduction : elle remplace les différentes occurrences d'une même variable par des termes cohérents. Si on se donne un terme t et qu'on identifie, parmi les sous-termes de t , un groupe de termes cohérents t_1, \dots, t_n , alors en remplaçant ces termes par une même variable x dans t on obtient un terme s , qui contient toutes les occurrences de x et tel qu'il existe un contexte C tel que $C[\lambda x s] = t$ et $C[\langle \lambda x s \rangle [t_1 \dots t_n]]$ contient t parmi ses réduits par construction. On a donc trouvé un moyen d'obtenir un parent de t . Or il faudrait un moyen de les obtenir tous. Remarque : le groupe de terme cohérents choisi ne peut pas être quelconque. Il y a une condition essentielle qu'il doit respecter, c'est que tous ces éléments doivent être des sous-termes libres (définition §3.1.28). Cette condition est relative, c'est à dire qu'elle dépend de la position du λ . Concrètement on doit d'abord placer le λ et ensuite choisir un groupe de sous-termes libres.

On peut agir sur deux paramètres :

- Pour un groupe de termes cohérents t_1, \dots, t_n fixé qui ont été remplacés par les occurrences d'une variable, on peut modifier la position

du λ qui lie cette variable dans le terme. Pour autant que ces termes restent des sous-termes libres de t .

- Pour une position de λ fixée on peut modifier le groupe de sous-termes libres cohérents que l'on choisit.

La solution adoptée est la suivante : on considère toutes les positions possible pour le λ et ensuite pour chaque position on considère tous les groupes de sous-termes libres cohérents qui sont sous ce λ . On doit définir une fonction qui calcule $\text{Sub}_L(t, \alpha)$, puis en extraire tous les ensembles de termes cohérents. Autrement dit toutes les cliques. Seulement il faut respecter certaines règles. Deux termes qui sont à la même adresse sont forcément dans la même clique (cf §3.1.30). Sinon lors de la β -expansion on aura, à la même adresse, une variable et un terme qui n'est pas cohérent avec cette variable. Donc la β -expansion ne sera pas uniforme et ce n'est pas ce qu'on veut. Donc on ne peut pas séparer deux termes qui sont à la même adresse. En calculant tous ces cas on obtient tous les parents en un étape de β -expansion. Il ne reste plus qu'à itérer le processus sur chaque terme obtenu pour avoir tous les parents d'un terme donné en un nombre d'étape de β -expansion fixé.

5.3 Cousins

Afin de comprendre un peu mieux le problème des termes cousins, on a écrit une fonction qui utilise la précédente et qui cherche un parent commun aux termes qu'on lui donne. On commence par calculer le ξ de chaque terme et vérifier que c'est bien le même, ce qui permet d'éviter beaucoup de calculs inutiles. Ensuite on fixe une borne de recherche n . Cette borne détermine le nombre de β -expansions que l'on va faire sur les termes de départ pour tenter de trouver un parent commun.

Soit t_1, t_2 deux termes dont on voudrait savoir si ils sont cousins. On commence par calculer $\text{Exp}(t_1)$ et $\text{Exp}(t_2)$ selon la méthode citée plus haut et on s'intéresse à l'ensemble $\text{Exp}(t_1) \cap \text{Exp}(t_2)$. Si cet ensemble n'est pas vide alors tous ses éléments sont des preuves qu'ils sont cousins. Si cet ensemble est vide alors on calcule $\bigcup_{s \in \text{Exp}(t_1)} \text{Exp}(s)$ et $\bigcup_{s \in \text{Exp}(t_2)} \text{Exp}(s)$ et on regarde leur intersection et on itère ce processus. À chaque fois qu'on fait une étape de β -expansion en plus, on incrémente un compteur. Lorsque le compteur à atteint la borne que l'on avait fixée au départ l'algorithme s'arrête et renvoie une valeur négative (si on atteint la borne c'est qu'on n'a pas trouvé de parent commun). Cet algorithme fonctionne avec des termes qui ne contiennent pas de monôme vide car ils ont la même taille et donc si ils ont un parent commun c'est en un même nombre d'étape de β -expansion. Le problème c'est

que le nombre de termes grandit assez vite et la borne que l'on fixe est difficilement plus grande que 5 pour un temps de calcul raisonnable (plusieurs heures) pour l'instant.

On peut raffiner un peu cet algorithme pour qu'il soit plus rapide. Au lieu de calculer l'intersection de tous les parents en k étapes de t_1 et de t_2 , on peut faire la chose suivante. On sait qu'à chaque élément de $\text{Exp}(t_1)$ correspond un unique élément de $\text{Exp}(t_2)$ qui lui est cohérent (dans le cas où t_1 et t_2 ont une forme normale). On les regroupe par paires et on relance l'algorithme sur chacune de ces paires. Si on considère qu'un élément de $\text{Exp}(t_1)$ est une "direction", le premier algorithme consiste à chercher dans toutes les directions en même temps, étape par étape, alors que le raffinement cherche dans une direction jusqu'à la borne, puis change de direction en cas d'échec. Dans le pire des cas ça revient au même, mais en moyenne le deuxième est plus efficace.

5.4 Correspondance entre la β -réduction et la Γ -réduction

Pour illustrer le théorème §3.2.18 on a écrit un programme qui le réalise. Tout comme la preuve de cette extension du résultat de Ehrhard-Regnier est complètement différente, le programme en lui-même (qui est une extension du programme écrit par Laurent Regnier et qui reprend en partie ce qu'il a fait) est assez différent de celui existant. Il permet notamment de traiter les cas qui ne terminent pas et qui ne pouvaient pas être calculés par l'ancien programme. Typiquement $(\Delta)\Delta$ donnait une erreur avec le programme de Laurent Regnier, alors que ce n'est pas le cas ici.

Pour arriver à ce résultat on a dû d'abord écrire une fonction qui permet de faire un étape de β -réduction pour le λ -calcul pur en choisissant le redex que l'on veut réduire. On itérera cette fonction autant que souhaité jusqu'à la forme normale, si elle existe. C'est la partie gauche du diagramme 3.2. Pour écrire cette fonction on a utilisé le même principe que pour la réduction dans le λ -calcul avec ressources. C'est à peu près la même fonction que celle qui fait la réduction pour le λ -calcul avec ressources, à la différence près que comme on ne calcule qu'un réduit, on n'a pas besoin de "marquer" les occurrences de la variable de substitution.

Une fois que l'on a réduit suffisamment notre λ -terme de départ, on peut donner un approximant ou laisser le programme en choisir un aléatoirement. On a donc écrit une fonction qui permet de générer aléatoirement un approximant d'un λ -terme donné. Cette fonction est calquée sur la définition

des approximatants. Concrètement on compare le λ -terme pur avec le λ -terme avec ressources et, comme ils sont construits de la même façon (par définition), pour chaque abstraction dans le λ -terme pur on donne le même nom à la variable liée dans le λ -calcul avec ressources.

On va alors reconstruire les parents successifs de l'approximant choisi en suivant la preuve de §3.2.18. La fonction qui fait la β -réduction retourne comme résultat non seulement le réduit mais en plus l'adresse du redex qui à été réduit ainsi que l'adresse des différentes occurrences de la variable de substitution. On commence par se servir de l'adresse du redex pour l'isoler. On commence avec un terme t de la forme $C[s]$ et on travaille avec s . On a écrit une fonction *subterm_extract* qui prend en argument le terme t et une liste d'adresses et qui renvoie un couple composé du terme s et du monôme des sous-termes aux adresses listées en argument. Cette liste contient les adresses auxquelles on veut remplacer le sous-terme qui s'y trouve par une variable, on utilisera "X" pour qu'il n'y ait pas de confusion avec les variables déjà utilisées. Cette fonction *subterm_extract* lit le terme s jusqu'à se trouver à l'endroit de la première adresse dans la liste. Elle mémorise le sous-terme qui s'y trouve dans une liste et le remplace par "X". Ensuite elle continue de même avec les autres adresses. Et finit par retourner (s, T) .

La fonction qui reconstruit effectivement le parent utilise *subterm_extract* et retourne le terme $C[\langle \lambda X s' \rangle T]$ où s' est le terme s dans lequel certains sous-termes ont été remplacés par "X". Ce terme est un parent de $C[s]$ de manière évidente.

Comme on "ajoute" un lambda devant s' on doit faire la même manipulation sur les indices de de Bruijn que pour la réduction, mais dans l'autre sens. En effet on doit, cette fois, incrémenter les indices et non pas les décrémenter.

Le programme que nous avons écrit nous permet de réduire autant de fois que souhaité le terme de départ, donc pour reconstruire le parent du terme d'arrivée on itère simplement ce processus en construisant tous les parents intermédiaires et en les affichant.

Chapitre 6

Conclusion

Dans ce travail nous avons défini une nouvelle réduction pour le λ -calcul avec ressources qui permet d'avoir une correspondance entre la β -réduction et celle-ci. Ainsi nous avons pu généraliser un résultat important de [ER06]. Ce résultat n'était valable que pour les termes en forme normale et nous avons pu le démontrer pour n'importe quels termes et ceci avec des méthodes différentes. Nous avons donc dû définir de nouveaux outils qui nous ont permis d'obtenir ce résultat. De plus nous avons écrit un programme (en OCaml) qui permet de réaliser effectivement le théorème démontré.

On a également défini une théorie de l'anti-réduction, appelée β -expansion, pour le λ -calcul avec ressources où le nombre de parents en une étape d'un terme est fini. Ces outils pourraient servir à démontrer certains résultats pour les termes cousins, mais n'ont pas encore été utilisés par manque de maîtrise de ces outils.

D'autre part, et malgré les faibles résultats, nous avons étudiés le lien qui existe entre les différents termes issus d'une même réduction (les cousins). Nous avons pu déterminer certaines conditions nécessaires pour que plusieurs termes soient cousins. Néanmoins aucune condition suffisante n'a pu être trouvée.

6.1 Travaux futurs

Ce travail sur les termes cousins a mis à jour beaucoup de questions qui n'ont pas pu être résolues ici.

On ne sait toujours pas si la relation de cousins est transitive. Soit t_1, t_2, t_3 trois termes avec ressources tels que t_1 soit cousin avec t_2 et t_3 de parent commun respectif s_2 et s_3 . En général s_2 et s_3 ne sont pas cohérents. Par

contre on sait qu'il existe au moins un terme s'_3 qui est un parent de t_3 et qui est cohérent avec s_2 , et de même il existe s'_2 qui est cohérent avec s_3 . On peut essayer de prouver la transitivité en montrant que soit s_2 et s'_3 sont cousins, soit que s_3 et s'_2 sont cousins. Ce qui n'est pas plus facile que de montrer que t_2 et t_3 sont cousins. Il faudrait pour ça plus d'informations sur les termes cousins.

On a vu dans le chapitre sur les relations familiales que pour un terme t qui ne contenait pas de λ ni de monôme vide, l'ensemble $\mathcal{F}(t)$ est fini. Comme on ne sait pas si la relation de cousins est transitive, on ne sait pas non plus si cet ensemble est une clique pour cette relation. Par contre comme $\mathcal{F}(t)$ est fini, on peut s'intéresser à une des plus grosses cliques qu'il contient (elle n'est pas forcément unique). Notons la $\mathcal{F}_{max}(t)$. On peut étudier la taille de cette clique par rapport à la taille de $\mathcal{F}(t)$. Mais on peut aussi s'intéresser aux parents de cette clique et notamment au plus petit, c'est à dire au plus proche parent qui se réduit en $\mathcal{F}_{max}(t)$. Ce terme a une particularité : tous les réduits de ses parents, à part lui, se réduisent en \emptyset . En effet, supposons que ça ne soit pas le cas. Alors il existe t' en forme normale qui n'appartient pas à $\mathcal{F}_{max}(t)$, mais qui est un cousin de chaque élément de $\mathcal{F}_{max}(t)$ par définition. C'est contradictoire.

Pour chaque clique de cousins on peut définir cette notion de plus proche parent (il n'y a pas nécessairement unicité). Ce plus proche parent, au sens de la réduction, est en quelque sorte le terme le plus petit qui contient toutes les informations que les termes de la clique partagent. On pourrait étudier la distance entre la clique et son plus proche parent par rapport à la taille de la clique.

Toutefois la transitivité de la relation de cousins n'entraînerait pas nécessairement l'égalité suivante : $\mathcal{F}_{max}(t) = \mathcal{F}(t)$. On pourrait envisager d'avoir trois termes qui sont cousins deux à deux mais qui, pourtant, n'ont pas de parent commun à tous les trois.

On a montré que la relation de cousins était semi-décidable. Pour avoir un résultat de décidabilité on peut essayer de borner la distance (en étape de réduction) entre deux termes et un éventuel parent commun. On devrait pouvoir définir une fonction de "différence" entre deux termes qui donne une borne maximale pour la distance entre les termes et un éventuel parent commun. Une autre méthode, dans le cas où pour un terme t , $\mathcal{F}(t)$ est fini, consisterait à étudier les termes qui se réduisent en un minimum d'étapes en un sous-ensemble maximum de $\mathcal{F}(t)$. Dans le cas où $\mathcal{F}(t)$ est fini cette notion est bien définie. On pourrait trouver des propriétés particulières à ces termes et ainsi espérer avoir une preuve de la décidabilité. C'est ce qui

nous fait penser que la relation de cousins serait décidable, au moins dans certains cas où les termes ne contiennent pas de monômes vides. Une dernière méthode consisterait à pouvoir caractériser complètement les termes cousins et la décidabilité en serait une conséquence directe.

Si on réussit à montrer la décidabilité de cette relation on peut se demander si on peut avoir une preuve constructive.

Nous avons déterminé certaines propriétés des termes cousins mais uniquement dans des cas particuliers (sans monôme vide et sans abstraction ni monôme vide). Le cas général est assez compliqué comme le peuvent en témoigner les exemples donnés dans la section 4.7. On ne sait pas si $\mathcal{F}(t)$ est fini lorsque t contient des λ , car ses cousins peuvent ne pas avoir la même taille que t . On ne sait pas non plus si étant donné t , un terme qui contient des monômes vides, on peut borner la taille de ses cousins.

Nous avons défini la β -expansion dans le but d'avoir une autre approche des termes cousins et espérer pouvoir prouver certaines propriétés par ce biais. Nous n'avons pas pu le faire par manque de maîtrise de cette nouvelle notion. En l'étudiant de manière plus approfondie on devrait pouvoir l'utiliser pour avoir des preuves différentes de certains résultats que nous avons déjà montrés et peut être réussir à démontrer de nouvelles choses grâce à cela.

Bibliographie

- [Bar84] Henk Barendregt. *The lambda-calculus : its syntax and semantics*. Elsevier, 1984.
- [BHP13] Pierre Boudes, Fanny He, and Michele Pagani. A characterization of the Taylor expansion of lambda-terms. In *Proceedings of the 22nd EACSL Annual Conference Computer Science Logic CSL 2013, September 2-5, 2013, Torino, Italy, 2013*.
- [Bou93] Gérard Boudol. The lambda-calculus with multiplicities. *Technical report 2025*, 1993.
- [Ehr02] Thomas Ehrhard. On Köthe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 2002.
- [Ehr10] Thomas Ehrhard. A finiteness structure on resource terms. *LICS*, 2010.
- [ER04] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1–3) :1–41, 2004.
- [ER06] Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 2006. To appear.
- [GBL99] Pierre-Louis Curien Gérard Boudol and Carolina Lavatelli. A semantics for lambda calculi with resource. *Mathematical Structures in Computer Science*, 9 :437–482, 1999.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [Kfo00] Assaf J. Kfoury. A linearization of the lambda-calculus. *Journal of Logic and Computations*, 10 :411–436, 2000.
- [Kri90] Jean-Louis Krivine. *Lambda-calcul : types et modèles*. Masson, 1990.
- [PRDR10] Michele Pagani and Simona Ronchi Della Rocca. Solvability in Resource Lambda-Calculus. In Luke Ong, editor, *Foundations of*

Software Science and Computational Structures, 13th International Conference (FOSSACS 2010), volume 6014 of *Lecture Notes in Comput. Sci.*, pages 358–373, 2010.

- [PT09] Michele Pagani and Paolo Tranquilli. Parallel Reduction in Resource Lambda-Calculus. In Zhenjiang Hu, editor, *Programming Languages and Systems, 7th Asian Symposium (APLAS 2009)*, volume 5904 of *Lecture Notes in Comput. Sci.*, pages 226–242, 2009.

Résumé de la thèse : Le lambda-calcul avec ressources est une variante du lambda-calcul fondée sur la linéarité : les lambda-termes avec ressources sont aux lambda-termes ce que sont les polynômes aux fonctions réelles, c'est à dire des approximations multi-linéaires. En particulier les réductions dans le lambda-calcul avec ressources peuvent être vues comme des approximations des beta-réductions, mais la contrainte de linéarité a des conséquences importantes, notamment la forte normalisation de la réduction avec ressources. Pour ainsi dire, la beta-réduction est obtenue par passage à la limite des réductions avec ressources qui l'approximent.

Cette thèse étudie les aspects combinatoires, très riches, du lambda-calcul avec ressources. On commence par définir précisément la notion de réduction avec ressource associée à une beta-réduction : étant donné un lambda-terme t , un approximant s de celui-ci et t' une beta-réduction de t , on lui associe une réduction avec ressources (appelée gamma-réduction) de s qui réduit les «mêmes» redex que celle de t et produit un ensemble S' d'approximants de t' . Cette définition permet de retrouver une preuve légèrement plus intuitive de l'un des théorèmes fondamentaux de la théorie, qui permet également de le généraliser.

Dans un second temps on étudie les relations «familiales» entre termes avec ressources, la question centrale étant de caractériser le fait que deux termes avec ressources sont des réduits d'un même terme. Ce problème central et difficile n'est pas pleinement résolu, mais la thèse présente plusieurs résultats préliminaires et développe les bases d'une théorie pour arriver à cette fin.