

# Un Lambda-Calcul Atomique

T. Gundersen<sup>1</sup>, W. Heijltjes<sup>2</sup>, & M. Parigot<sup>3</sup>

1: *Laboratoire Preuves, Programmes, Systèmes, CNRS & Université Paris Diderot*

teg@jklm.no

2: *INRIA & Laboratoire d'Informatique de l'École Polytechnique*

willem.heijltjes@gmail.com

3: *Laboratoire Preuves, Programmes, Systèmes, CNRS & Université Paris Diderot*

parigot@pps.univ-paris-diderot.fr

## Résumé

Nous introduisons un lambda-calcul avec partage explicite, le *lambda-calcul atomique*, dans lequel la duplication des sous-termes est faite pas à pas en fonction des constructeurs. Nous donnons une fonction de dénotation du lambda-calcul atomique dans le lambda-calcul et montrons que le lambda-calcul atomique simule la  $\beta$ -réduction et préserve la normalisation forte. Nous donnons aussi un système de type pour le lambda-calcul atomique et montrons que la réduction préserve le type.

## 1. Introduction

Le  $\lambda$ -calcul est un modèle de calcul dans lequel on ne sait pas, en général, maîtriser l'usage des ressources. Même dans le cas des  $\lambda$ -termes simplement typés, un chemin de réduction peut terminer en temps linéaire, alors qu'un autre peut avoir une croissance hyperexponentielle. Le point crucial pour maîtriser cette croissance est d'exercer un contrôle des duplications et effacements: calculer à l'intérieur des termes avant de les dupliquer et éviter des calculs inutiles à l'intérieur de termes qui seront effacés. Une façon de le faire consiste à utiliser des *stratégies de réduction* dans le but de choisir les meilleurs chemin de réduction. Une autre consiste à adapter la syntaxe, comme le font les *calculs de substitution explicite* [ACCL91] et les *réductions de graphe optimales* [Lam90].

Dans cet article nous considérons un aspect particulier du contrôle des duplications: le caractère *atomique* de la duplication. Nous voulons dire par là: la possibilité de dupliquer chaque constructeur individuellement; en particulier, étant donné un terme  $\lambda x.t$ , il s'agit de dupliquer l'abstraction  $\lambda x$  indépendamment du corps  $t$  du terme. Nous présentons un calcul qui implémente un telle réduction, le *lambda-calcul atomique*, inspiré par l'élimination des coupures en *inférence profonde* [Gug07] – un système de déduction où les règles d'inférence s'appliquent à n'importe quelle sous-formule, à la manière de la réécriture de termes. L'étape de duplication cruciale est la suivante, représentée ici dans un formalisme appelé *open deduction*, où les connecteurs s'appliquent à la fois aux formules et aux preuves [GGP10].

$$\begin{array}{c}
 A^x \rightarrow \frac{A^x}{\parallel_t^B} \\
 \hline
 [A \rightarrow B] \wedge [A \rightarrow B]
 \end{array}
 \rightsquigarrow
 \begin{array}{c}
 A^x \rightarrow \frac{A^x}{\parallel_t^B} \\
 \hline
 \frac{c}{B \wedge B} \\
 \hline
 [A \rightarrow B] \wedge [A \rightarrow B]
 \end{array}$$

Les déductions ci-dessus est constituée comme suit. Dans la déduction à gauche, la double ligne verticale de  $A$  à  $B$  représente une déduction correspondant au terme  $t$  avec une variable libre de type  $A$ , qui est combinée avec une implication pour produire une déduction correspondant au terme  $\lambda x.t : A \rightarrow B$ . A cette déduction est appliquée une *contraction* ( $c$ ) qui permet d'utiliser sa conclusion  $A \rightarrow B$  deux fois dans une autre déduction. A droite, au lieu de dupliquer en entier la sous-déduction correspondant à  $\lambda x.t$ , la contraction est déplacée

à l'intérieur de la sous-déduction qui est le conséquent de l'implication, en utilisant une règle spéciale de *distribution* ( $d$ ). La contraction ( $c$ ) peut alors continuer à dupliquer  $t$  par étapes jusqu'à atteindre le sommet ( $A$ ), après quoi la distribution ( $d$ ) peut être remplacée par la conjonction des deux déductions correspondant à  $\lambda x.t$ .

La caractéristique principale du lambda-calcul atomique sera précisément un constructeur de termes, le *distributeur*, correspondant à la règle de distribution précédente. Dans la suite nous définirons les termes du lambda-calcul atomique et ses règles de réduction et nous en établirons les propriétés de base: systèmes de type simple, traduction entre lambda-calcul atomique et lambda-calcul, et préservation de la normalisation forte. Nous n'utiliserons pas plus avant dans cet article le formalisme de l'open deduction car il est possible de donner un système de déduction de type pour le lambda-calcul atomique, sous la forme plus familière de calcul de séquents. Pour aider l'intuition, nous donnerons une représentation des constructeurs et des règles de réduction en termes de graphes. Les questions plus spécialisées, telles que les stratégies de réduction et leur efficacité, seront abordées dans des travaux ultérieurs, en particulier les relations avec les travaux sur le partage paresseux [Wad71], [BLM07], [Bal12]. Dans la limite des connaissances des auteurs, il ne semble pas qu'il existe d'autre calcul de termes qui possède la propriété d'atomicité, propriété qui est en revanche centrale dans les réductions de graphe optimales.

## 2. Les lambda-termes atomiques

Le lambda-calcul atomique et le lambda-calcul standard utiliseront deux alphabets distincts pour éviter les confusions. Le lambda-calcul standard ( $\Lambda$ ) est défini ainsi:

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

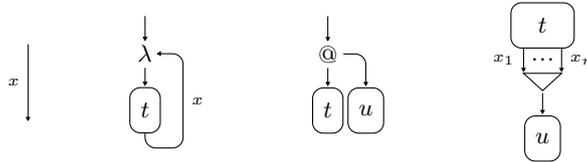
La base du lambda-calcul atomique est un lambda-calcul linéaire avec des contractions explicites. Dans la suite, nous supposons que toutes les variables introduites sont nouvelles, chaque fois que cela s'applique.

**Définition 1.** Le calcul de base  $\Lambda_A^-$  est défini par la grammaire suivante:

$$s, t, u, v, w \quad := \quad x \quad | \quad \lambda x.t \quad | \quad (t)u \quad | \quad t[x_1, \dots, x_n \leftarrow u]$$

où (i) chaque variable apparaît au plus une fois, (ii) dans  $\lambda x.t$ , la variable  $x$  doit apparaître dans  $t$  et devient liée, et (iii) dans  $t[x_1, \dots, x_n \leftarrow u]$ , chaque variable  $x_i$  doit apparaître dans  $t$  et devient liée.

Le constructeur  $t[x_1, \dots, x_n \leftarrow u]$  est appelé un *partage* et un partage d'arité zéro,  $t[\leftarrow u]$ , est appelé un *affaiblissement*. Les quatre constructeurs (*variable*, *abstraction*, *application*, et *partage*) sont représentés graphiquement comme suit. Dans la suite, les boîtes telles que celles entourant  $t$  et  $u$  pourront être omises.



La fonction  $(-)_A : \Lambda \rightarrow \Lambda_A^-$  envoie les lambda-termes dans les termes de base. La traduction consiste à remplacer inductivement un terme  $\lambda x.N$  par  $\lambda x.N'[x_1, \dots, x_n \leftarrow x]$ , où  $N'$  est obtenu à partir de  $N$  en remplaçant les  $n$  occurrences de  $x$  par  $x_1, \dots, x_n$ . Pour définir formellement la traduction, celle-ci est paramétrée par une application  $\sigma$  qui assigne à chaque variable  $x$  du terme, une suite de nouvelles variables  $x_1, \dots, x_n$  dont la longueur  $n$  est le nombre d'occurrences de  $x$  dans le terme, et telle que les variables apparaissant dans l'image de  $\sigma$  sont toutes distinctes. Nous désignerons par  $|N|_x$  le nombre d'occurrences de  $x$  dans  $N$ .

Le terme  $\langle N \rangle_\sigma$  est défini inductivement par

$$\begin{aligned} \langle x \rangle_\sigma &= \sigma(x) \\ \langle (M)N \rangle_\sigma &= \langle (M) \rangle_{\sigma'} \langle N \rangle_{\sigma''} \\ \langle \lambda x.N \rangle_\sigma &= \begin{cases} \lambda x'. \langle N \rangle_\sigma & \text{si } |N|_x = 1 \text{ et } \sigma(x) = x' \\ \lambda x. \langle N \rangle_\sigma[x_1, \dots, x_n \leftarrow x] & \text{si } |N|_x = n \neq 1 \text{ et } \sigma(x) = (x_1, \dots, x_n), \end{cases} \end{aligned}$$

où  $\sigma'$  et  $\sigma''$  sont construites comme suit: pour chaque variable  $x$  de  $(M)N$  avec  $|M|_x = m$  et  $|N|_x = n$ , si  $\sigma(x) = (x_1, \dots, x_{m+n})$ , alors  $\sigma'(x) = (x_1, \dots, x_m)$  et  $\sigma''(x) = (x_{m+1}, \dots, x_{m+n})$ . Remarquons que dans le cas d'une abstraction linéaire  $\lambda x.t$ , où  $x$  a une seule occurrence dans  $t$ , la traduction n'introduit pas de partage  $[x_1 \leftarrow x]$ . Ceci permettra la réécriture d'un partage unaire  $t[x \leftarrow u]$  en utilisant une substitution linéaire de  $u$  à  $x$ .

Les termes figurant dans l'image de  $\langle - \rangle$  sont ceux du calcul de base engendrés par la grammaire:

$$t := x \mid \lambda x.t \mid (s)t \mid \lambda x.t[x_1, \dots, x_n \leftarrow x] \quad (n \neq 1).$$

**Définition 2.** Le *lambda-calcul atomique*  $\Lambda_A$  étend le calcul de base avec un nouveau constructeur, le *distributeur*. On définit en même temps par des grammaires mutuellement récursives, les *termes*  $t$  et les *termes de multiplicité*  $n$ , dénotés  $t^n$ , pour chaque  $n > 0$ .

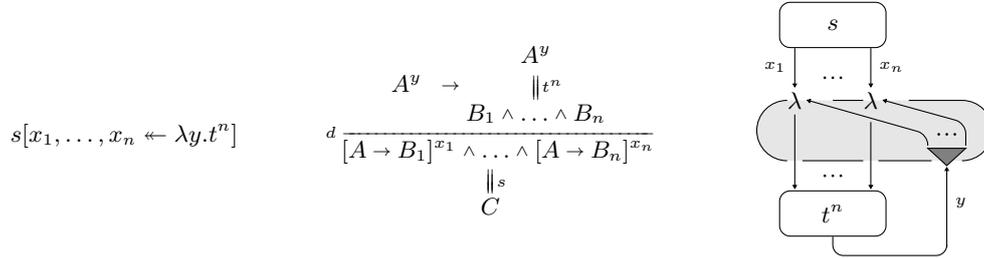
$$s, t, u, v, w := \dots \mid s[x_1, \dots, x_n \leftarrow \lambda y.t^n]$$

$$t^n := \langle t_1, \dots, t_n \rangle \mid t^n[x_1, \dots, x_m \leftarrow u] \mid t^n[x_1, \dots, x_m \leftarrow \lambda y.t^m]$$

où (i) chaque variable apparaît au plus une fois, (ii) dans  $s[x_1, \dots, x_n \leftarrow \lambda y.t^n]$ , chaque  $x_i$  doit apparaître dans  $s$  et devient lié, et (iii) dans  $t^n[x_1, \dots, x_n \leftarrow u]$  et  $t^n[x_1, \dots, x_m \leftarrow \lambda y.t^m]$ , chaque  $x_i$  doit apparaître dans  $t^n$  et devient lié.

Les deux constructeurs, partage et distributeur, seront désignés ensemble comme *clôtures*. Les termes lambda-calcul atomique sont appelés *lambda-termes atomiques*. Un terme de multiplicité  $n$  n'est rien d'autre qu'un  $n$ -uple de lambda-termes atomiques auquel des clôtures sont appliquées comme s'il s'agissait d'un lambda-terme atomique. Dit autrement, un terme de multiplicité  $n$  est de la forme  $\langle t_1, \dots, t_n \rangle[\Gamma]$  où  $[\Gamma]$  est une suite de partages et distributeurs.

Remarquons que, contrairement aux clôtures qui s'appliquent aussi bien aux lambda-termes atomiques et aux termes de multiplicité quelconque, l'application et l'abstraction ne s'appliquent qu'aux lambda-termes atomiques proprement dits. Dans la suite, la distinction entre les termes et les termes de multiplicité  $n$  sera faite seulement quand cela est pertinent, et les deux seront en général désignés par les mêmes lettres  $t$  ( $u, v, \dots$ ). Le distributeur est représenté ci-dessous sous forme de terme, de déduction et de graphe.



La déduction, ci-dessus au centre, est construite à partir des éléments suivants: une sous-déduction  $t^n$  dont la conclusion est une conjonction  $n$ -aire; une implication à partir de  $A$  qui correspond à l'abstraction  $\lambda y$  à l'intérieur du distributeur dans le terme à gauche; une règle de distributeur  $d$  qui permet de déduire  $[A \rightarrow B_1]^{x_1} \wedge \dots \wedge [A \rightarrow B_n]^{x_n}$  de  $A \rightarrow [B_1 \wedge \dots \wedge B_n]$ ; une sous-déduction  $s$  dont les hypothèses  $A \rightarrow B_i$  correspondent aux variables  $x_i$  liée par le distributeur dans le terme  $s$  à gauche. La représentation graphique du distributeur est tournée de 180 degrés par rapport à la déduction; graphiquement, le fait que  $t^n$  soit de multiplicité  $n$  signifie qu'il a  $n$  liens entrants. Comme le suggère la représentation graphique, le distributeur contient un noeud de *dé-partage* ou *co-partage* implicite, du type de ceux utilisés dans les

graphes de réduction optimale. Le problème crucial de la réduction de graphe, résolu pour la première fois dans [Lam90], est de décider, quand un noeud de partage rencontre un noeud de co-partage, s'ils se dupliquent entre eux ou se neutralisent. Dans le lambda-calcul atomique, le distributeur contient le co-partage et transporte son propre champ, ce qui rend le problème de la décision triviale. Cependant, cela signifie que les noeuds pour le lambda contenus dans la boîte du distributeur, contrairement au cas de la réduction de graphe optimale, ne peuvent pas participer à un redex. Actuellement aucune autre façon de faire, compatible avec un système de type, n'est connue: en inférence profonde, la manière naturelle d'implémenter une  $\beta$ -réduction comportant une abstraction implicite d'un distributeur conduit à des règles de déduction incorrectes. Ceci est cohérent avec la conviction généralement partagée selon laquelle il pourrait être impossible d'avoir un système de type pour la réduction de graphe optimale.

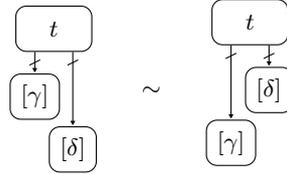
Dans le maniement des clôtures, nous supposons certaines équivalences naturelles entre termes. Nous commençons par étendre et formaliser la notation  $[\Gamma]$ . Soit  $[\gamma]$  une clôture (un partage ou un distributeur) vu comme opération sur les termes, prenant un argument à gauche,  $t[\gamma]$ , et soit  $[\Gamma]$  une suite de telles opérations, qui peut être vue comme une opération composée sur les termes,  $t[\Gamma]$ . La structure naturelle de  $[\Gamma]$  est celle d'un ensemble partiellement ordonné, où  $[x_1, \dots, x_n \leftarrow s] < [y_1, \dots, y_m \leftarrow t]$  si une quelconque des variables liées  $y_i$  du second partage a une occurrence libre dans le corps de la première,  $s$ . On définit alors l'ensemble des *variables libres*  $\text{FV}[\gamma]$  et des *variables liées*  $\text{BV}[\gamma]$  d'une clôture comme suit.

$$\begin{aligned} \text{FV}[x_1, \dots, x_n \leftarrow t] &= \text{FV}(t) & \text{BV}[x_1, \dots, x_n \leftarrow t] &= \{x_1, \dots, x_n\} \\ \text{FV}[x_1, \dots, x_n \leftarrow \lambda y.t] &= \text{FV}(\lambda y.t) & \text{BV}[x_1, \dots, x_n \leftarrow \lambda y.t] &= \{x_1, \dots, x_n\} \end{aligned}$$

Les lambda-termes atomiques seront considérés modulo l'équivalence (1).

$$t[\gamma][\delta] \sim t[\delta][\gamma] \quad \text{si} \quad \text{FV}[\gamma] \cap \text{BV}[\delta] = \emptyset \quad (1)$$

Dit autrement, si  $[\gamma] < [\delta]$  alors les clôtures  $[\gamma]$  et  $[\delta]$  doivent être appliquées dans l'ordre  $t[\gamma][\delta]$  et sinon elles peuvent être permutées. L'équivalence (1) est illustrée ci-dessous. Une flèche barrée ( $\downarrow$ ) représente un ensemble de liens et est utilisée à la place d'une ellipse quand l'espace est limité.



La fonction  $\llbracket - \rrbracket : \Lambda_A \rightarrow \Lambda$ , appelée *dénotation*, envoie le lambda-calcul atomique dans le lambda-calcul. Les vraies substitutions (de  $t$  à  $x$  dans  $s$ ) sont dénotées  $s\{t/x\}$ .

$$\begin{aligned} \llbracket x \rrbracket &= x \\ \llbracket \lambda x.u \rrbracket &= \lambda x.\llbracket u \rrbracket \\ \llbracket (u)t \rrbracket &= (\llbracket u \rrbracket)\llbracket t \rrbracket \\ \llbracket u[x_1, \dots, x_n \leftarrow t] \rrbracket &= \llbracket u \rrbracket\{t/x_1\} \dots \{t/x_n\} \\ \llbracket u[x_1, \dots, x_n \leftarrow \lambda y.\langle t_1, \dots, t_n \rangle[\Gamma]] \rrbracket &= \llbracket u \rrbracket\{\lambda y.\llbracket t_1[\Gamma] \rrbracket/x_1\} \dots \{\lambda y.\llbracket t_n[\Gamma] \rrbracket/x_n\} \end{aligned}$$

Dans le dernier cas, dans  $t_i[\Gamma]$  les clôtures figurant dans  $[\Gamma]$  peuvent contenir des variables liées qui n'apparaissent pas dans  $t_i$ , ce qui signifie que la condition de linéarité est, au sens strict, violée: ceci ne pose pas de problème car les clôtures dans  $[\Gamma]$  sont transformées immédiatement en vraies substitutions.

**Proposition 3.** *Pour tous lambda-termes atomiques  $u$  et  $v$  tels que  $u \sim v$ , on a  $\llbracket u \rrbracket = \llbracket v \rrbracket$ .*

*Preuve.* Simple vérification de chacun des cas.  $\square$

### 3. Réductions de partage

Le but des nouveaux constructeurs, partage et distributeur, est d'implémenter la duplication atomique. Bien qu'il y ait des ressemblances évidentes dans la notation et la conception entre le lambda-calcul atomique et les calculs de substitution explicite, l'objectif est en un certain sens orthogonal à celui des substitutions explicites, qui est d'exercer un contrôle plus fin du processus de substitution dans un calcul de termes. La différence d'objectif se manifeste dans les règles de réduction de partage, qui sont présentées dans cette section.

**Définition 4.** La relation  $\rightsquigarrow_S$  est la relation de réécriture sur  $\Lambda_A$  induite par les règles de réduction (2)–(12).

Le premier ensemble de règles de réduction, (2)–(7) ci-dessous, déplace les partages et distributeurs en direction de la racine de l'arbre du terme, un point d'attache canonique pour les partages et distributeurs, permettant la mise en oeuvre des autres règles de réduction en particulier la règle (12). La direction de réécriture est exactement l'opposé de celle des calculs de substitution explicite, dans lesquels les substitutions explicites se déplacent en direction des feuilles de l'arbre du terme. De plus, les règles de réduction (2)–(7) sont des identités dans la représentation graphique, alors que les règles de réécriture des calculs de substitution explicite ne le sont pas car elles peuvent dupliquer des clôtures.

$$\lambda x.t[\gamma] \rightsquigarrow_S (\lambda x.t)[\gamma] \quad \text{si } x \notin \text{FV}[\gamma] \quad (2)$$

$$(u[\gamma])t \rightsquigarrow_S ((u)t)[\gamma] \quad (3)$$

$$(u)t[\gamma] \rightsquigarrow_S ((u)t)[\gamma] \quad (4)$$

$$\langle t_1, \dots, t_i[\gamma], \dots, t_n \rangle \rightsquigarrow_S \langle t_1, \dots, t_n \rangle[\gamma] \quad (5)$$

$$u[x_1, \dots, x_n \leftarrow t[\gamma]] \rightsquigarrow_S u[x_1, \dots, x_n \leftarrow t][\gamma] \quad (6)$$

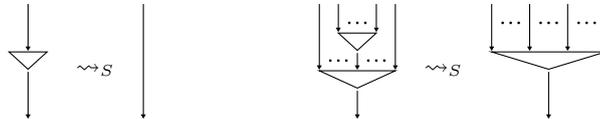
$$u[x_1, \dots, x_n \leftarrow \lambda y.t[\gamma]] \rightsquigarrow_S u[x_1, \dots, x_n \leftarrow \lambda y.t][\gamma] \quad \text{si } y \notin \text{FV}[\gamma] \quad (7)$$

Un second ensemble de règles de réécriture concerne l'opérateur de partage pris séparément. Les règles ci-dessous, transforment le partage unaire en vraie substitution et composent des partages consécutifs.

$$u[x \leftarrow t] \rightsquigarrow_S u\{t/x\} \quad (8)$$

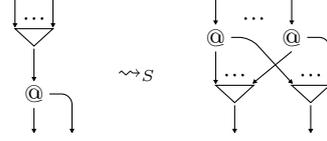
$$u[x_1, \dots, x_n \leftarrow y_i][y_1, \dots, y_i, \dots, y_m \leftarrow t] \rightsquigarrow_S u[y_1, \dots, y_{i-1}, x_1, \dots, x_n, y_{i+1}, \dots, y_m \leftarrow t] \quad (9)$$

Remarquons que bien que les partages unaires, objets de la règle (8), n'apparaissent pas dans les termes directement traduits du lambda-calcul, ils peuvent être créés par exemple en combinant un partage binaire et un partage d'arité zéro dans la règle (9). Les deux règles de réécriture sont représentées graphiquement ci-dessous.

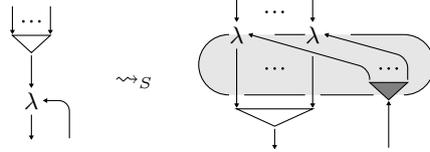


Le troisième ensemble de règles de réécriture, (10)–(12), concerne la duplication effective des termes. Il s'agit des étapes de duplication atomique qui constituent le coeur du calcul. Une représentation graphique est donnée en dessous de chaque règle. Ensemble, elles permettent de voir pourquoi les règles de réduction sont (presque) exhaustives: dans (10), un partage agit sur une application; dans (11) un partage agit sur une abstraction; et dans (12) un partage rencontre un co-partage à la fin du champ d'un distributeur. Manque le cas où un partage rencontre un distributeur; ce cas est redondant et a été omis à cause de la complexité des termes impliqués (voir Section 7 pour plus d'explications).

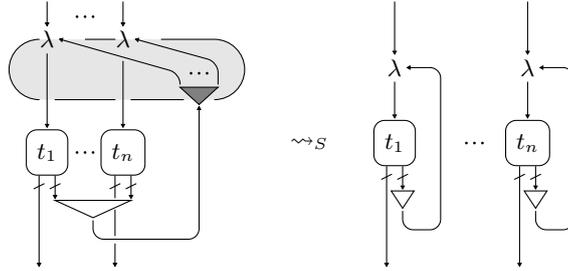
$$u[x_1, \dots, x_n \leftarrow (v)t] \rightsquigarrow_S u\{(y_1)z_1/x_1\} \dots \{(y_n)z_n/x_n\}[y_1, \dots, y_n \leftarrow v][z_1, \dots, z_n \leftarrow t] \quad (10)$$



$$u[x_1, \dots, x_n \leftarrow \lambda x.t] \rightsquigarrow_S u[x_1, \dots, x_n \leftarrow \lambda x.\langle y_1, \dots, y_n \rangle [y_1, \dots, y_n \leftarrow t]] \quad (11)$$



$$\begin{aligned} u[x_1, \dots, x_n \leftarrow \lambda y.\langle t_1, \dots, t_n \rangle [y_{1,1}, \dots, y_{1,k_1}, \dots, y_{n,1}, \dots, y_{n,k_n} \leftarrow y]] &\rightsquigarrow_S \\ u\{\lambda y_{1,1}.t_1 [y_{1,1}, \dots, y_{1,k_1} \leftarrow y_1] / x_1\} \dots \{\lambda y_{n,1}.t_n [y_{n,1}, \dots, y_{n,k_n} \leftarrow y_n] / x_n\} & \\ \text{où } \{y_{i,1}, \dots, y_{i,k_i}\} \subseteq \text{FV}(t_i) \text{ pour tout } i \leq n & \quad (12) \end{aligned}$$



Une remarque mineure concernant cette règle: un distributeur de la forme présente à gauche ci-dessous, où  $y$  à une unique occurrence dans  $t_i$ , peut être réduit comme s'il était de la forme présente à droite ci-dessous

$$u[x_1, \dots, x_n \leftarrow \lambda y.\langle t_1, \dots, t_n \rangle] \sim u[x_1, \dots, x_n \leftarrow \lambda y'.\langle t_1, \dots, t_n \rangle [y \leftarrow y']]$$

Il est intéressant de considérer les instances d'arité zéro des trois règles précédentes prises séparément.

$$\begin{aligned} t[\leftarrow (u)v] &\rightsquigarrow_S t[\leftarrow u][\leftarrow v] \\ t[\leftarrow \lambda x.u] &\rightsquigarrow_S t[\leftarrow \lambda x.\langle \rangle][\leftarrow u] \\ t[\leftarrow \lambda y.\langle \rangle][\leftarrow y] &\rightsquigarrow_S t \end{aligned}$$

Avec les autres règles, elles permettent de d'effectuer des réductions du type suivant.

$$t[\leftarrow u] \rightsquigarrow_S^* t[\leftarrow x_1] \dots [\leftarrow x_n] \quad \text{où } \{x_1, \dots, x_n\} = \text{FV}(u)$$

Les principales propriétés des réductions de partage ( $\rightsquigarrow_S$ ) sont données ci-dessous.

**Proposition 5.** *La réduction  $\rightsquigarrow_S$  satisfait la propriété de normalisation forte.*

La preuve de la Proposition 5 est reportée à la Section 5, car elle plus facile à formuler avec les notations introduites dans cette section. Les lambda-termes atomiques en forme normale relativement à la réduction  $\rightsquigarrow_S$  sont dits être en *forme normale de partage*. Ils correspondent essentiellement aux traductions de lambda-termes standards par la fonction  $(-)$  – la correspondance ne vaut strictement que pour les termes clos car il faut exclure les termes ayant des variables libres dans des affaiblissements.

**Proposition 6.** *Les lambda-termes atomiques clos en forme normale de partage sont précisément l'image de l'ensemble des lambda-termes clos par  $(-)$ . De plus, les traductions  $(-)$  et  $\llbracket - \rrbracket$  sont inverses l'une de l'autre pour les termes clos en forme normale de partage.*

*Preuve.* Simple vérification de chacun des cas.  $\square$

**Proposition 7.** *Pour tous lambda-termes atomiques  $u$  and  $v$ , si  $u \rightsquigarrow_S v$ , alors  $\llbracket u \rrbracket = \llbracket v \rrbracket$ .*

*Preuve.* Simple vérification de chacun des cas.  $\square$

**Théorème 8.** *La réduction  $\rightsquigarrow_S$  possède les propriétés de normalisation forte et de confluence.*

*Preuve.* La réduction  $\rightsquigarrow_S$  possède la propriété de normalisation forte par la Proposition 5; par la Proposition 7 la dénotation est préservée par réduction et par la Proposition 6 les formes normales sont en bijection avec les dénotations (bien que la proposition ne vaille que pour les termes clos, le résultat vaut en général en prenant simplement la clôture des termes); par conséquent  $\rightsquigarrow_S$  est confluente.  $\square$

## 4. Typage du lambda-calcul atomique

Le lambda-calcul atomique typé est défini par les règles ci-dessous. Comme pour le lambda-calcul atomique nous avons deux sortes d'expressions. D'une part des *jugements pour les lambda-termes atomiques*,  $x_1 : A_1, \dots, x_n : A_n \vdash t : B$ , où  $t$  est un lambda-terme atomique,  $A_1, \dots, A_n, B$  des formules formées à partir de  $\rightarrow$ , et  $x_1, \dots, x_n$ , des variables distinctes qui sont exactement celles qui apparaissent dans  $t$ . D'autre part des *jugements pour les termes de multiplicité  $k > 0$* ,  $x_1 : A_1, \dots, x_n : A_n \vdash t^k : B_1 \wedge \dots \wedge B_k$ , où  $t^k$  est un terme de multiplicité  $k$ ,  $A_1, \dots, A_n, B_1, \dots, B_k$  des formules formées à partir de  $\rightarrow$ , et  $x_1, \dots, x_n$  des variables distinctes qui sont exactement celles qui apparaissent dans  $t^k$ . Les antécédents  $x_1 : A_1, \dots, x_n : A_n$  des jugements sont considérés comme des ensembles et sont notés  $\Gamma$  or  $\Delta$ . En outre  $x_1 : B_1, \dots, x_n : B_n$  sera abrégé en  $x_1 \dots x_n : B$ .

**Règles de typage pour les lambda-termes atomiques:**

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x. t : A \rightarrow B} \lambda \qquad \frac{\Gamma \vdash u : A \rightarrow B \quad \Delta \vdash v : A}{\Gamma, \Delta \vdash (u)v : B} @$$

$$\frac{}{x : A \vdash x : A} \text{ax} \qquad \frac{\Gamma, x_1 \dots x_n : B \vdash u : A \quad \Delta \vdash t : B}{\Gamma, \Delta \vdash u[x_1, \dots, x_n \leftarrow t] : A} \leftarrow$$

$$\frac{\Gamma, x_1 : A \rightarrow B_1, \dots, x_n : A \rightarrow B_n \vdash s : C \quad \Delta, y : A \vdash t^n : B_1 \wedge \dots \wedge B_n}{\Gamma, \Delta \vdash s[x_1, \dots, x_n \leftarrow \lambda y. t^n] : C} \leftarrow \leftarrow$$

**Règles de typage pour les termes de multiplicité  $k$ :**

$$\frac{\Gamma_1 \vdash t_1 : A_1 \quad \dots \quad \Gamma_k \vdash t_k : A_k}{\Gamma_1, \dots, \Gamma_k \vdash \langle t_1, \dots, t_k \rangle : A_1 \wedge \dots \wedge A_k} \langle \rangle_k$$

$$\frac{\Gamma, x_1 \dots x_n : B \vdash u^k : A_1 \wedge \dots \wedge A_k \quad \Delta \vdash t : B}{\Gamma, \Delta \vdash u^k[x_1, \dots, x_n \leftarrow t] : A_1 \wedge \dots \wedge A_k} \leftarrow k$$

$$\frac{\Gamma, x_1 : A \rightarrow B_1, \dots, x_n : A \rightarrow B_n \vdash u^k : C_1 \wedge \dots \wedge C_k \quad \Delta, y : A \vdash t^n : B_1 \wedge \dots \wedge B_n}{\Gamma, \Delta \vdash u^k[x_1, \dots, x_n \leftarrow \lambda y. t^n] : C_1 \wedge \dots \wedge C_k} \leftarrow \leftarrow k$$

$$\begin{array}{c}
 \frac{\overline{x_1 : C \vdash x_1 : C}^{\text{ax}} \quad \dots \quad \overline{x_n : C \vdash x_n : C}^{\text{ax}}}{\Gamma, x_1 \dots x_n : C \vdash u : A} \pi_1 \quad \frac{\Delta \vdash v : B \rightarrow C \quad \Sigma \vdash t : B}{\Delta, \Sigma \vdash (v)t : C} \textcircled{\text{A}}}{\Gamma, \Delta, \Sigma \vdash u[x_1, \dots, x_n \leftarrow (v)t] : A} \leftarrow \\
 \zeta_S \\
 \frac{\overline{y_1 : B \rightarrow C \vdash y_1 : B \rightarrow C}^{\text{ax}} \quad \overline{z_1 : B \vdash z_1 : B}^{\text{ax}} \quad \dots \quad \overline{y_n : B \rightarrow C \vdash y_n : B \rightarrow C}^{\text{ax}} \quad \overline{z_n : B \vdash z_n : B}^{\text{ax}}}{z_1 : B, y_1 : B \rightarrow C \vdash (y_1)z_1 : C} \textcircled{\text{A}} \quad \dots \quad \frac{\overline{y_n : B \rightarrow C \vdash y_n : B \rightarrow C}^{\text{ax}} \quad \overline{z_n : B \vdash z_n : B}^{\text{ax}}}{z_n : B, y_n : B \rightarrow C \vdash (y_n)z_n : C} \textcircled{\text{A}}}{\Gamma, z_1 \dots z_n : B, y_1 \dots y_n : B \rightarrow C \vdash u\{(y_1)z_1/x_1\} \dots \{(y_n)z_n/x_n\} : A} \pi'_1 \quad \frac{\Delta \vdash v : B \rightarrow C}{\Delta, \Sigma \vdash (v)t : C} \leftarrow \quad \frac{\Sigma \vdash t : B}{\Sigma \vdash t : B} \leftarrow}{\Gamma, \Delta, \Sigma \vdash u\{(y_1)z_1/x_1\} \dots \{(y_n)z_n/x_n\}[y_1, \dots, y_n \leftarrow v] : A} \leftarrow \quad \frac{\Gamma, \Delta, \Sigma \vdash u\{(y_1)z_1/x_1\} \dots \{(y_n)z_n/x_n\}[y_1, \dots, y_n \leftarrow v][z_1, \dots, z_n \leftarrow t] : A}{\Gamma, \Delta, \Sigma \vdash u\{(y_1)z_1/x_1\} \dots \{(y_n)z_n/x_n\}[y_1, \dots, y_n \leftarrow v][z_1, \dots, z_n \leftarrow t] : A} \leftarrow
 \end{array}$$

Figure 1: Dédutions correspondant à la règle de réduction (10) dans la preuve de la Proposition 9

$$\begin{array}{c}
 \frac{\Gamma, x_1 \dots x_n : B \rightarrow C \vdash u : A \quad \frac{\Delta, x : B \vdash t : C}{\Delta \vdash \lambda x.t : B \rightarrow C} \lambda}{\Gamma, \Delta \vdash u[x_1, \dots, x_n \leftarrow \lambda x.t] : A} \leftarrow \\
 \zeta_S \\
 \frac{\Gamma, x_1 \dots x_n : B \rightarrow C \vdash u : A \quad \frac{\overline{y_1 : C \vdash y_1 : C}^{\text{ax}} \quad \dots \quad \overline{y_n : C \vdash y_n : C}^{\text{ax}}}{y_1 \dots y_n : C \vdash \langle y_1, \dots, y_n \rangle : (C \wedge \dots \wedge C)} \langle \rangle \quad \frac{\Delta, x : B \vdash t : C}{\Delta, x : B \vdash t : C} \leftarrow}{\Gamma, \Delta \vdash u[x_1, \dots, x_n \leftarrow \lambda x.\langle y_1, \dots, y_n \rangle[y_1, \dots, y_n \leftarrow t]] : A} \leftarrow
 \end{array}$$

Figure 2: Dédutions correspondant à la règle de réduction (11) dans la preuve de la Proposition 9

$$\begin{array}{c}
 \frac{\Gamma, x_1 \dots x_n : B \rightarrow C \vdash u : A \quad \frac{\Delta_1, y_{1,1} \dots y_{1,k_1} : B \vdash t_1 : C \quad \dots \quad \Delta_n, y_{n,1} \dots y_{n,k_n} : B \vdash t_n : C}{\Delta_1, \dots, \Delta_n, y_{1,1} \dots y_{n,k_n} : B \vdash \langle t_1, \dots, t_n \rangle : (C \wedge \dots \wedge C)} \langle \rangle \quad \frac{y : B \vdash y : B}{y : B \vdash y : B}^{\text{ax}}}{\Gamma, \Delta_1, \dots, \Delta_n \vdash u[x_1, \dots, x_n \leftarrow \lambda y.\langle t_1, \dots, t_n \rangle[y_{1,1}, \dots, y_{1,k_1}, \dots, y_{n,1}, \dots, y_{n,k_n} \leftarrow y]] : A} \leftarrow \\
 \zeta_S \\
 \frac{\frac{\Delta_1, y_{1,1} \dots y_{1,k_1} : B \vdash t_1 : C \quad \overline{y_1 : B \vdash y_1 : B}^{\text{ax}}}{\Delta_1, y_1 : B \vdash t_1[y_{1,1}, \dots, y_{1,k_1} \leftarrow y_1] : C} \leftarrow \quad \frac{\Delta_n, y_{n,1} \dots y_{n,k_n} : B \vdash t_n : C \quad \overline{y_n : B \vdash y_n : B}^{\text{ax}}}{\Delta_n, y_n : B \vdash t_n[y_{n,1}, \dots, y_{n,k_n} \leftarrow y_n] : C} \leftarrow}{\Delta_1 \vdash \lambda y_1.t_1[y_{1,1}, \dots, y_{1,k_1} \leftarrow y_1] : B \rightarrow C} \lambda \quad \dots \quad \frac{\Delta_n \vdash \lambda y_n.t_n[y_{n,1}, \dots, y_{n,k_n} \leftarrow y_n] : B \rightarrow C}{\Delta_n \vdash \lambda y_n.t_n[y_{n,1}, \dots, y_{n,k_n} \leftarrow y_n] : B \rightarrow C} \lambda}{\Gamma, \Delta_1, \dots, \Delta_n \vdash u\{\lambda y_1.t_1[y_{1,1}, \dots, y_{1,k_1} \leftarrow y_1]/x_1\} \dots \{\lambda y_n.t_n[y_{n,1}, \dots, y_{n,k_n} \leftarrow y_n]/x_n\} : A} \pi'
 \end{array}$$

Figure 3: Dédutions correspondant à la règle de réduction (12) dans la preuve de la Proposition 9

**Proposition 9.** *Les réductions de partage préservent le type.*

*Preuve.* Parmi les règles de réduction (2)–(12) seules les étapes de duplication, (10)–(12), seront considérées. Les autres cas sont analogues, mais plus simples.

Pour la règle de réduction (10) les déductions de type correspondant aux termes sont données dans la Figure 1. Dans la deuxième déduction,  $\pi'_1$  est obtenue à partir de  $\pi_1$  en remplaçant chaque axiome  $x_i : C \vdash x_i : C$  par une preuve de  $z_i : B, y_i : B \rightarrow C \vdash (y_i)z_i : C$ .

Pour la règle (11) les déductions sont données dans la Figure 2.

Pour la règle (12) les déductions sont données dans la Figure 3. Dans la deuxième déduction,  $\pi'$  est obtenue à partir de  $\pi$  en remplaçant chaque axiome  $x_i : B \rightarrow C \vdash x_i : B \rightarrow C$  par une preuve de  $\Delta_i \vdash \lambda y_i. t_i [y_{i,1}, \dots, y_{i,k_i} \leftarrow y_i] : B \rightarrow C$ .  $\square$

Les propositions suivantes sont faciles à vérifier:

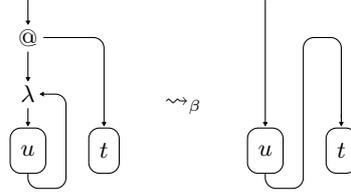
**Proposition 10.** *Pour tous lambda-termes atomiques  $u$  et  $v$  tels que  $u \sim v$ , si  $v : A$  alors  $u : A$ .*

**Proposition 11.** *Pour tout lambda-terme  $M$ , si  $M : A$  alors  $(M) : A$ .*

## 5. Bêta-réduction

La définition de la  $\beta$ -réduction dans le lambda-calcul atomique est identique à celle du lambda-calcul usuel. Elle est définie et représentée graphiquement ci-dessous.

$$(\lambda x.u)t \rightsquigarrow_{\beta} u\{t/x\}$$



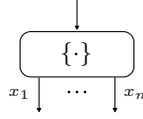
Cependant, son effet est très différent: dans le lambda-calcul atomique, la  $\beta$ -réduction est une opération linéaire, car la variable liée  $x$  a exactement une occurrence dans le corps du terme,  $u$ . La duplication du terme  $t$  dans le lambda-calcul atomique se fait au travers des réductions de partage ( $\rightsquigarrow_S$ ).

**Théorème 12.** *La  $\beta$ -réduction préserve le type.*

*Preuve.* La transformation des déductions correspondant à la  $\beta$ -réduction est la suivante, où  $\pi'_1$  est obtenue à partir de  $\pi_1$  en remplaçant chaque axiome  $x : A \vdash x : A$  par la déduction  $\pi_2$  de  $\Delta \vdash t : A$ .

$$\frac{\begin{array}{c} \cdots \quad \overline{x : A \vdash x : A} \quad \cdots \\ \pi_1 \\ \Gamma, x : A \vdash u : B \\ \hline \Gamma \vdash \lambda x.u : A \rightarrow B \end{array} \quad \begin{array}{c} \pi_2 \\ \Delta \vdash t : A \end{array}}{\Gamma, \Delta \vdash (\lambda x.u)t : B} \rightsquigarrow_{\beta} \frac{\begin{array}{c} \pi_2 \\ \cdots \quad \Delta \vdash t : A \quad \cdots \\ \pi'_1 \\ \Gamma, \Delta \vdash u\{t/x\} : B \end{array}}{\Gamma, \Delta \vdash u\{t/x\} : B} \quad \square$$

Pour relier les étapes de réduction du lambda-calcul atomique à celles du lambda-calcul, nous introduisons une notion de *contexte à un trou* pour les lambda-termes atomiques. Le trou, représenté par  $\{\cdot\}$ , peut être vu comme un sous-terme unique spécial, indiquant une certaine position dans le terme. Pour pouvoir être utilisés dans des clôtures, les trous seront indexés par des suites de variables  $x_1, \dots, x_n$  – les variables libres du trou. Graphiquement, cela peut être représenté par un ensemble de liens sortants, comme suit.



**Définition 13.** Les *contextes atomiques*  $\Lambda_A^{\{\cdot\}}$  sont construits par la grammaire des lambda-termes atomiques étendue au constructeur de *trou*  $\{\cdot\}_{x_1, \dots, x_n}$ , dans lequel les variables  $x_1, \dots, x_n$  sont considérées comme libres, et doivent avoir une unique occurrence dans le contexte atomique.

$$t\{\cdot\}_{x_1, \dots, x_n} := \dots \mid \{\cdot\}_{x_1, \dots, x_n}$$

Le lambda-terme atomique  $t\{u\}$  est obtenu à partir du contexte atomique  $t\{\cdot\}_{x_1, \dots, x_n}$  en substituant  $u$  au trou, sous la condition que  $\text{FV}(u) = \{x_1, \dots, x_n\}$ .

L'indice indiquant les variables libres d'un trou sera omis à chaque fois que cela est possible. Le rôle du trou dans un lambda-terme atomique  $t\{u\}$  est d'identifier un sous-terme spécifique  $u$ , et en particulier d'obtenir des traductions indépendantes de  $t\{\cdot\}$  et  $u$  dans le lambda-calcul. Il y a deux détails importants dans cette traduction. Premièrement, puisque le trou peut se trouver dans un sous-terme qui est partagé, un contexte correspondant dans le lambda-calcul peut avoir besoin d'un nombre quelconque de trous. Deuxièmement, une variable libre d'un trou dans un contexte atomique peut être liée par une clôture  $[\gamma]$  (mais aussi par une abstraction  $\lambda x$ ). Puisque dans la traduction, l'action de  $[\gamma]$  devient naturellement une substitution, la traduction d'un contexte atomique  $t\{\cdot\}_{x_1, \dots, x_n}$  sera un couple  $(N\{\cdot\}, \sigma)$  formé d'un lambda-contexte standard (défini ci-dessous) et d'une substitution  $\sigma$  qui remplace les variables  $x_1, \dots, x_n$  par des lambda-termes. Alors, si le contexte atomique  $t\{\cdot\}_{x_1, \dots, x_n}$  se traduit par  $(N\{\cdot\}, \sigma)$  et le lambda-terme atomique  $u$  avec variables libres  $\{x_1, \dots, x_n\}$  se traduit par  $M$ , la traduction de  $t\{u\}$  sera  $N\{M\sigma\}$ , i.e. le contexte  $N\{\cdot\}$  dont les trous sont remplis par le lambda-terme obtenu en appliquant  $\sigma$  à  $M$ .

**Définition 14.** *Lambda-contextes*  $\Lambda^{\{\cdot\}}$  étend le lambda-calcul standard  $\Lambda$  avec un constructeur de *trou*  $\{\cdot\}$ .

$$N\{\cdot\} := \dots \mid \{\cdot\}$$

Pour les lambda-contextes il n'y a aucune restriction sur le nombre d'occurrences du trou et le trou n'a pas de variables libres.

**Définition 15.** La dénotation  $\llbracket u\{\cdot\}_{x_1, \dots, x_n} \rrbracket$  d'un contexte atomique est un couple  $(M\{\cdot\}, \sigma)$ , où  $M\{\cdot\} \in \Lambda^{\{\cdot\}}$  et  $\sigma: \{x_1, \dots, x_n\} \rightarrow \Lambda$  est une substitution, définie comme suit.

Dans la suite, soit  $\llbracket u\{\cdot\} \rrbracket = (M\{\cdot\}, \sigma)$ , et soit  $\llbracket v\{\cdot\}[\Gamma] \rrbracket = (N\{\cdot\}, \tau)$ .

$$\llbracket \{\cdot\}_{x_1, \dots, x_n} \rrbracket = (\{\cdot\}, \emptyset)$$

$$\llbracket \lambda x. u\{\cdot\} \rrbracket = (\lambda x. M\{\cdot\}, \sigma)$$

$$\llbracket (u\{\cdot\})t \rrbracket = ((M\{\cdot\})\llbracket t \rrbracket, \sigma)$$

$$\llbracket (t)u\{\cdot\} \rrbracket = ((\llbracket t \rrbracket)M\{\cdot\}, \sigma)$$

$$\llbracket t[y_1, \dots, y_m \leftarrow u\{\cdot\}] \rrbracket = (\llbracket t \rrbracket\{M\{\cdot\}/y_1\} \dots \{M\{\cdot\}/y_m\}, \sigma)$$

$$\begin{aligned} \llbracket u\{\cdot\}[y_1, \dots, y_m \leftarrow t] \rrbracket &= (M\{\cdot\}\{\llbracket t \rrbracket/y_1\} \dots \{\llbracket t \rrbracket/y_m\}, \\ &\quad \sigma\{\llbracket t \rrbracket/y_1\} \dots \{\llbracket t \rrbracket/y_m\} \upharpoonright_{\{x_1, \dots, x_n\}}) \end{aligned}$$

$$\begin{aligned} \llbracket t[y_1, \dots, y_m \leftarrow \lambda z. \langle t_1, \dots, v\{\cdot\}, \dots, t_m \rangle[\Gamma]] \rrbracket &= (\llbracket t \rrbracket\{N_1/y_1\} \dots \{\lambda z. N\{\cdot\}/y_i\} \dots \{N_m/y_m\}, \sigma) \\ &\quad \text{où } N_i = \lambda z. \llbracket t_i[\Gamma] \rrbracket \end{aligned}$$

$$\begin{aligned} \llbracket u\{\cdot\}[y_1, \dots, y_m \leftarrow \lambda z. \langle t_1, \dots, t_m \rangle[\Gamma]] \rrbracket &= (M\{\cdot\}\{N_1/y_1\} \dots \{N_m/y_m\}, \\ &\quad \sigma\{N_1/y_1\} \dots \{N_m/y_m\} \upharpoonright_{\{x_1, \dots, x_n\}}) \\ &\quad \text{où } N_i = \lambda z. \llbracket t_i[\Gamma] \rrbracket \end{aligned}$$

**Lemme 16.** Si  $\llbracket u\{\cdot\}_{x_1, \dots, x_n} \rrbracket = (M\{\cdot\}, \sigma)$  et  $\text{FV}(t) = \{x_1, \dots, x_n\}$  alors  $\llbracket u\{t\} \rrbracket = M\{\llbracket t \rrbracket\sigma\}$ .

*Preuve.* Simple vérification de chacun des cas.  $\square$

Une première utilisation des contextes atomiques est la preuve de la proposition 5, qui avait été reportée en Section 3.

**Proposition 5 (de la section 3).** *La réduction  $\rightsquigarrow_S$  vérifie la propriété de normalisation forte.*

*Preuve.* On associe à chaque clôture  $[\gamma]$  d'un lambda-terme atomique  $u$  deux mesures: la *profondeur*, qui est la distance du sous-terme à la racine du terme quand celui-ci est représenté par un graphe dirigé acyclique modulo l'équation (1); et le *poids*, qui est  $1/2$  pour un distributeur et qui, pour un partage, est la taille de la dénotation du terme partagé, qui est définie comme suit. Considérons un partage  $[\gamma] = [x_1, \dots, x_n \leftarrow t]$  dans un terme  $u$  avec  $u = s\{t\}$  et  $\llbracket u\{\cdot\} \rrbracket = (N\{\cdot\}, \sigma)$ . Le poids de  $[\gamma]$  est alors la taille de  $\llbracket t \rrbracket \sigma$ , mesurée par le nombre d'abstractions et d'applications. Remarquons que les deux mesures sont invariantes par l'équation (1).

La mesure d'un terme  $t$  est le couple (poids, profondeurs) formé du multiensemble des poids de toutes les clôtures de  $t$ , et du multiensemble des profondeurs de toutes les clôtures. L'examen de chacun des cas montre que pour toute réduction de partage, on a l'une des situations suivantes:

- la profondeur d'une clôture  $[\delta]$  diminue, alors que tous les poids et toutes les autres profondeurs sont inchangés (règles de réduction (2)–(7));
- un partage est supprimé, alors que les poids de tous les autres restent inchangés (règles de réduction (8)–(9));
- une ou plusieurs clôtures  $[\delta]$  sont remplacées par d'autres de poids strictement plus petit: dans la règle de réduction (10), un partage de poids  $n$  est remplacé par deux partages de poids  $n - 1$ ; dans la règle de réduction (11) un partage de poids  $n \geq 1$  est remplacé par un distributeur de poids  $1/2$  et un partage de poids  $n - 1$ ; et dans la règle de réduction (12) un distributeur de poids  $1/2$  et un partage de poids zéro sont remplacés par plusieurs partages de poids zéro.

Il s'en suit que la réduction  $\rightsquigarrow_S$  fait décroître strictement la mesures des termes, ce qui prouve la proposition.  $\square$

Une étape de  $\beta$ -réduction dans le lambda-calcul atomique correspond à  $n \geq 0$  étapes de  $\beta$ -réduction dans le lambda-calcul standard.

**Lemme 17.** *Pour tous lambda-termes atomiques  $u$  et  $v$ , si  $u \rightsquigarrow_\beta v$  alors  $\llbracket u \rrbracket \rightsquigarrow_\beta^* \llbracket v \rrbracket$ .*

*Preuve.* Un lambda-terme atomique avec un  $\beta$ -redex est de la forme  $v\{(\lambda x.u)t\}$ . Soit  $\llbracket v\{\cdot\} \rrbracket = (M\{\cdot\}, \sigma)$ ; alors en utilisant le lemma 16,

$$\llbracket v\{(\lambda x.u)t\} \rrbracket = M\{((\lambda x.\llbracket u \rrbracket)\llbracket t \rrbracket)\sigma\} \rightsquigarrow_\beta^n M\{(\llbracket u \rrbracket\{\llbracket t \rrbracket/x\})\sigma\} = \llbracket v\{u\{t/x\}\} \rrbracket$$

où  $n$  est le nombre de trous dans  $M\{\cdot\}$ .  $\square$

De plus, chaque étape de  $\beta$ -réduction au niveau de la dénotation d'un lambda-terme atomique peut être simulée dans le lambda-calcul atomique par la composition de réductions de partage et d'une  $\beta$ -réduction.

**Théorème 18.** *Si  $\llbracket u \rrbracket \rightsquigarrow_\beta M$  alors il existe des lambda-termes atomiques  $v$  et  $t$  tels que  $u \rightsquigarrow_S^* v \rightsquigarrow_\beta t$  et  $\llbracket t \rrbracket = M$ .*

*Preuve.* Prendre pour  $v$  la forme normale de partage de  $u$ .  $\square$

## 6. Préservation de la normalisation forte

Nous montrons dans cette section que le lambda-calcul atomique préserve la normalisation forte relativement au lambda-calcul. La difficulté de la tâche, comme pour nombre de mécanismes de partage et calculs de substitution explicite, réside dans le fait que la réduction en lambda-calcul atomique peut se faire à l'intérieur d'un affaiblissement  $[\leftarrow t]$ . Les pas de  $\beta$ -réduction dans un affaiblissement sont simulés par zéro pas de  $\beta$ -réduction dans le lambda-terme correspondant, où les affaiblissements ne sont pas gardés. Ceci fait obstacle à la construction directe d'une réduction infinie en lambda-calcul à partir d'une réduction infinie en lambda-calcul atomique.

Afin de séparer le problème de la réduction des affaiblissements des détails des mécanismes de partage en lambda-calcul atomique, la preuve est scindée en deux étapes. Dans une première étape, on définit un *calcul d'affaiblissement explicite*, qui est un lambda-calcul avec affaiblissements construit pour fournir une dénotation adéquate du lambda-calcul atomique. On montre alors qu'un pas de  $\beta$ -réduction en lambda-calcul atomique correspond à au moins un pas de réduction dans le calcul d'affaiblissement explicite. Dans une seconde étape, on montre que le calcul d'affaiblissement explicite préserve la normalisation forte relativement au lambda-calcul et que donc le lambda-calcul atomique fait de même.

**Définition 19.** Le *calcul d'affaiblissement explicite*, appelé *w-calcul*, est défini par grammaire suivante.

$$S, T, U := x \mid \lambda x.T \text{ où } x \in \text{FV}(T) \mid (S)U \mid S[\leftarrow U] \mid \bullet$$

Les termes du w-calcul sont appelés *w-termes*. La  $\beta$ -réduction du w-calcul est standard:

$$(\lambda x.T)U \rightsquigarrow_{\beta} T\{U/x\}$$

La réduction des affaiblissements dans le w-calcul se fait de la manière suivante.

$$\begin{array}{ll} S[\leftarrow \lambda x.U] \rightsquigarrow_w S[\leftarrow U\{\bullet/x\}] & \lambda x.T[\leftarrow U] \rightsquigarrow_w (\lambda x.T)[\leftarrow U] \text{ (**)} \\ S[\leftarrow (T)U] \rightsquigarrow_w S[\leftarrow T][\leftarrow U] & (S[\leftarrow U])T \rightsquigarrow_w ((S)T)[\leftarrow U] \\ S[\leftarrow \bullet] \rightsquigarrow_w S & (S)T[\leftarrow U] \rightsquigarrow_w ((S)T)[\leftarrow U] \\ S[\leftarrow U] \rightsquigarrow_w S \text{ (*)} & S[\leftarrow T[\leftarrow U]] \rightsquigarrow_w S[\leftarrow T][\leftarrow U] \end{array}$$

\* si  $U$  est un sous-terme de  $S$

\*\* si  $x \notin \text{FV}(U)$

En tant qu'interprétation du lambda-calcul atomique, le w-calcul est *dénotationnel* vis à vis du partage, et *opérationnel* vis à vis de l'affaiblissement: le partage sera modélisé par la duplication (via la substitution), alors que la principale caractéristique du w-calcul est un mécanisme d'élimination des affaiblissements construit pour mimer celui du lambda-calcul atomique.

**Définition 20.** La fonction  $\llbracket - \rrbracket_w$  traduit les lambda-termes atomiques en w-termes.

$$\begin{aligned} \llbracket x \rrbracket_w &= x & \llbracket \lambda x.t \rrbracket_w &= \lambda x.\llbracket t \rrbracket_w & \llbracket t(u) \rrbracket_w &= (\llbracket t \rrbracket_w)\llbracket u \rrbracket_w \\ \llbracket t[x_1, \dots, x_n \leftarrow u] \rrbracket_w &= \begin{cases} \llbracket t \rrbracket_w[\leftarrow \llbracket u \rrbracket_w] & \text{si } n = 0 \\ \llbracket t \rrbracket_w\{\llbracket u \rrbracket_w/x_1\} \dots \{\llbracket u \rrbracket_w/x_n\} & \text{sinon} \end{cases} \\ \llbracket t[x_1, \dots, x_n \leftarrow \lambda y.\langle u_1, \dots, u_n \rangle[\Gamma]] \rrbracket_w &= \begin{cases} \llbracket t[\Gamma] \rrbracket_w\{\bullet/y\} & \text{si } n = 0 \\ \llbracket t \rrbracket_w\{\lambda y.\llbracket u_1[\Gamma] \rrbracket_w/x_1\} \dots \{\lambda y.\llbracket u_n[\Gamma] \rrbracket_w/x_n\} & \text{sinon} \end{cases} \end{aligned}$$

Dans la traduction du lambda-calcul atomique en w-calcul, les affaiblissements dans le champ d'un partage, par exemple  $[\leftarrow v]$  dans  $t[x, y \leftarrow u[\leftarrow v]]$ , sont dupliqués. Par contraste, la réduction en lambda-calcul atomique ne duplique jamais d'affaiblissements  $[\leftarrow v]$ . La réduction  $S[\leftarrow U] \rightsquigarrow_w S$  permet l'effacement des affaiblissements dupliqués et maintient ainsi la commutativité de la traduction et de la réduction. Le lemme suivant exprime ce fait.

**Lemme 21.** Si  $t \rightsquigarrow_{\beta} u$  alors  $\llbracket t \rrbracket_w \rightsquigarrow_{\beta}^+ \llbracket u \rrbracket_w$ . Si  $t \rightsquigarrow_S u$  alors  $\llbracket t \rrbracket_w \rightsquigarrow_w^* \llbracket u \rrbracket_w$ .

*Preuve.* Nous faisons d'abord la preuve pour des réductions à la racine d'un terme. Le cas de la  $\beta$ -réduction est traité ci-dessous; remarquons qu'ici,  $x$  a exactement une occurrence dans  $t$ , et au plus une dans  $\llbracket t \rrbracket_w$ .

$$\llbracket (\lambda x.t)u \rrbracket_w = (\lambda x.\llbracket t \rrbracket_w)\llbracket u \rrbracket_w \rightsquigarrow_{\beta} \llbracket t \rrbracket_w\{\llbracket u \rrbracket_w/x\} = \llbracket t\{u/x\} \rrbracket_w$$

Pour les permutations de partage (2)–(7), la plupart des cas sont faciles à vérifier. Le cas (2) (pour  $n \neq 0$ ) est traité explicitement ci-dessous; celui de (7) est analogue.

$$\begin{aligned} \llbracket t[x_1, \dots, x_n \leftarrow u[\leftarrow v]] \rrbracket_w &= \llbracket t \rrbracket_w\{\llbracket u \rrbracket_w[\leftarrow \llbracket v \rrbracket_w]/x_1\} \dots \{\llbracket u \rrbracket_w[\leftarrow \llbracket v \rrbracket_w]/x_n\} \\ &\rightsquigarrow_w^+ \llbracket t \rrbracket_w\{\llbracket u \rrbracket_w/x_1\} \dots \{\llbracket u \rrbracket_w/x_n\}\{\llbracket v \rrbracket_w\} \dots \{\leftarrow \llbracket v \rrbracket_w\} \\ &\rightsquigarrow_w^+ \llbracket t \rrbracket_w\{\llbracket u \rrbracket_w/x_1\} \dots \{\llbracket u \rrbracket_w/x_n\}\{\leftarrow \llbracket v \rrbracket_w\} \\ &= \llbracket t[x_1, \dots, x_n \leftarrow u][\leftarrow v] \rrbracket_w \end{aligned}$$

Dans la seconde étape de la transformation ci-dessus, les multiples copies de  $[\leftarrow \llbracket v \rrbracket_w]$  sont permutées en direction de la racine de  $\llbracket t \rrbracket_w$ . Ceci est possible car aucun  $\lambda y$  à l'intérieur de  $t$  ne peut être lié dans  $v$ , de par la structure du terme original  $t[x_1, \dots, x_n \leftarrow u[\leftarrow v]]$ . Dans la troisième étape les instances dupliquées de  $[\leftarrow \llbracket v \rrbracket_w]$  sont effacées.

Le cas de la règle (8) est immédiat d'après les égalités suivantes.

$$\llbracket u[x \leftarrow t] \rrbracket_w = \llbracket u \rrbracket_w\{\llbracket t \rrbracket_w/x\} = \llbracket u\{t/x\} \rrbracket_w$$

Pour la règle de réduction (9) il y a plusieurs sous-cas: pour  $n \neq 0$  le résultat est immédiat car les partages sont interprétés par des substitutions; pour  $n = 0$  et  $m > 1$  il découle des transformations ci-dessous, et pour  $n = 0$  et  $m = 1$  il s'obtient de la même manière.

$$\begin{aligned} \llbracket u[\leftarrow y_i][y_1, \dots, y_i, \dots, y_m \leftarrow t] \rrbracket_w &= \llbracket u \rrbracket_w[\leftarrow \llbracket t \rrbracket_w]\{\llbracket t \rrbracket_w/y_1\} \dots \{\llbracket t \rrbracket_w/y_m\} \\ &\rightsquigarrow_w \llbracket u \rrbracket_w\{\llbracket t \rrbracket_w/y_1\} \dots \{\llbracket t \rrbracket_w/y_m\} \\ &= \llbracket u[y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_m \leftarrow t] \rrbracket_w \end{aligned}$$

Remarquons que l'affaiblissement  $[\leftarrow \llbracket t \rrbracket_w]$  peut être supprimé par la règle  $S[\leftarrow U] \rightsquigarrow_w S$  car les substitutions  $\{\llbracket t \rrbracket_w/y_i\}$  assurent que  $\llbracket t \rrbracket_w$  apparaît dans le corps du terme.

Parmi les pas de duplication proprement dits, à savoir les règles (11)–(12), nous montrons en détail (11) et (12) et omettons (10). Pour la règle (11), il y a de nouveau deux cas. Dans le cas  $n = 0$  on a :

$$\begin{aligned} \llbracket u[\leftarrow \lambda x.t] \rrbracket_w &= \llbracket u \rrbracket_w[\leftarrow \lambda x.\llbracket t \rrbracket_w] \\ &\rightsquigarrow_w \llbracket u \rrbracket_w[\leftarrow \llbracket t \rrbracket_w\{\bullet/x\}] \\ &= \llbracket u[\leftarrow \lambda x.\langle \rangle][\leftarrow t] \rrbracket_w \end{aligned}$$

et dans le cas  $n \neq 0$  on a :

$$\begin{aligned} \llbracket u[x_1, \dots, x_n \leftarrow \lambda x.t] \rrbracket_w &= \llbracket u \rrbracket_w\{\lambda x.\llbracket t \rrbracket_w/x_1\} \dots \{\lambda x.\llbracket t \rrbracket_w/x_n\} \\ &= \llbracket u \rrbracket_w\{\lambda x.y_1\{\llbracket t \rrbracket_w/y_1\}/x_1\} \dots \{\lambda x.y_n\{\llbracket t \rrbracket_w/y_n\}/x_n\} \\ &= \llbracket u[x_1, \dots, x_n \leftarrow \lambda x.\langle y_1, \dots, y_n \rangle][y_1, \dots, y_n \leftarrow t] \rrbracket_w \end{aligned}$$

De manière analogue, pour la règle (12), dans le cas  $n = 0$  on a :

$$\llbracket u[\leftarrow \lambda y.\langle \rangle][\leftarrow y] \rrbracket_w = \llbracket u \rrbracket_w[\leftarrow \bullet] \rightsquigarrow_w \llbracket u \rrbracket_w$$

et dans le cas  $n \neq 0$  on a :

$$\begin{aligned} \llbracket u[x_1, \dots, x_n \leftarrow \lambda y.\langle t_1, \dots, t_n \rangle][y_{1,1}, \dots, y_{1,k_1}, \dots, y_{n,1}, \dots, y_{n,k_n} \leftarrow y] \rrbracket_w \\ &= \llbracket u \rrbracket_w\{T_1/x_1\} \dots \{T_n/x_n\} \quad \text{où } T_i = \lambda y_i.\llbracket t_i \rrbracket_w\{y/y_{i,1}\} \dots \{y/y_{i,k_i}\} \\ &=_{\alpha} \llbracket u\{t'_i/x_1\} \dots \{t'_n/x_n\} \rrbracket_w \quad \text{où } t'_i = \lambda y_i.t_i[y_{i,1}, \dots, y_{i,k_i} \leftarrow y_i] \end{aligned}$$

Enfin, pour les pas de réduction qui ne sont pas à la racine du terme, il est facile de voir que, puisque la traduction  $\llbracket - \rrbracket_w$  n'efface pas, si  $\llbracket t \rrbracket_w \rightsquigarrow \llbracket u \rrbracket_w$  alors  $\llbracket s\{t\} \rrbracket_w \rightsquigarrow^+ \llbracket s\{u\} \rrbracket_w$ .  $\square$

L'interprétation naturelle d'un w-terme par un lambda-terme, qui consiste à supprimer tous les affaiblissements  $[\leftarrow U]$ , est notée  $[-]$ . Cette interprétation a la propriété suivante, qui est facile à vérifier.

**Proposition 22.** *Pour tout lambda-terme atomique  $t$ ,  $\llbracket [t]_w \rrbracket = \llbracket t \rrbracket$ .*

Nous montrons que le w-calcul préserve la normalisation forte du lambda-calcul en utilisant la notion de *stratégie perpétuelle* [Bar84]. La principale propriété de cette stratégie de réduction est que si elle termine pour un terme donné, alors ce terme est fortement normalisable. Il suffit donc, pour montrer la préservation de la normalisation forte, de montrer que la stratégie perpétuelle infinie sur les w-termes se traduit en une suite de réductions infinie sur les lambda-termes.

**Définition 23.** La stratégie perpétuelle sur un w-terme  $U$  est une suite

$$U = U_1 \rightsquigarrow U_2 \rightsquigarrow U_3 \rightsquigarrow \dots$$

où  $U_{i+1} = \omega(U_i)$  est défini comme suit:

$$\omega(x) = x \qquad \omega(\lambda x.T) = \lambda x.\omega(T) \qquad \omega(T[\leftarrow U]) = \omega(T)[\leftarrow U]$$

$$\omega((T)U) = \begin{cases} (T')U[\leftarrow V] & \text{si } T = T'[\leftarrow V] \\ T'\{U/x\} & \text{si } T = \lambda x.T' \text{ et } x \in \text{FV}[T'] \\ T'\{U/x\} & \text{si } T = \lambda x.T' \text{ et } x \notin \text{FV}[T'] \text{ et } U \text{ est en forme normale} \\ (T)\omega(U) & \text{si } T = \lambda x.T' \text{ et } x \notin \text{FV}[T'] \text{ et } U \text{ n'est pas en forme normale} \\ (\omega(T))U & \text{sinon} \end{cases}$$

**Lemme 24.** *Si  $\omega(U\{S\}) = U\{\omega(S)\}$ , alors aucune suite de réduction partant de  $U\{S\}$  ne peut dupliquer  $S$  ni instancier de variables libres dans  $S$ . Dit plus formellement, si  $\omega(U\{S\}) = U\{\omega(S)\}$ , alors tout  $V$  tel que  $U\{S\} \rightsquigarrow V$  est de la forme  $V = U'\{S'\}$  avec  $U\{\bullet\} \rightsquigarrow U'\{\bullet\}$  et  $S \rightsquigarrow S'$ .*

*Preuve.* Par récurrence sur la définition de  $\omega$ . □

**Lemme 25.** *Si  $N$  est un lambda-terme, et  $U_1, U_2, U_3, \dots$  est une stratégie perpétuelle sur  $U_1 = \llbracket (N) \rrbracket_w$ , alors  $U_i = U_{i+1}$  si et seulement si  $U_i$  est en forme normale.*

*Preuve.* Remarquons que les seuls affaiblissements apparaissant dans  $U_1$  sont des variables et que tous les redex hors des affaiblissements sont réduits dans la stratégie perpétuelle. Par la définition de  $\omega$  et le lemme précédent, tous les affaiblissements apparaissant dans la stratégie perpétuelle de  $U_1$  sont en forme normale. □

**Lemme 26.** *Soient  $N$  un lambda-terme et  $i \geq 0$ . Si  $\omega^i(\llbracket (N) \rrbracket_w)$  a une suite de réductions infinie, alors la stratégie perpétuelle sur  $\omega^{i+1}(\llbracket (N) \rrbracket_w)$  est infinie.*

*Preuve.* Soit  $U_1, U_2, U_3, \dots$  la suite de réductions infinie partant de  $U_1 = \omega^i(\llbracket (N) \rrbracket_w)$  et soit  $S$  le plus petit sous-terme de  $U_1$  tel que  $\omega(U_1\{S\}) = U_1\{\omega(S)\}$ . Supposons que  $S = (T[\leftarrow W])V$  (les deux autres cas où  $S = (\lambda x.V)T$  se traitent de manière similaire). Il y a deux cas à considérer: le cas où le redex  $S$  est réduit dans la suite de réduction infinie, et le cas où il ne l'est pas.

- Si le redex  $S$  est réduit, alors nous avons par le lemme 24 une suite

$$U_1, U_2, \dots, U_k\{(T_k[\leftarrow W]_k)V_k\}, U_k\{(T_k)V_k[\leftarrow W]_k\}$$

avec  $U_1\{\bullet\} \rightsquigarrow U_k\{\bullet\}$ ,  $T \rightsquigarrow T_k$ ,  $V \rightsquigarrow V_k$  and  $W \rightsquigarrow W_k$ . Il s'en suit alors que  $U_1\{(T)V[\leftarrow W]\} \rightsquigarrow^* U_k\{(T_k)V_k[\leftarrow W]_k\}$ .

- Si le redex  $S$  n'est pas réduit, alors l'un des termes  $U_1\{\bullet\}$ ,  $T$ ,  $W$  ou  $V$  a une suite de réductions infinie par le lemme 24 et il en est donc de même pour  $U_1\{(T)V[\leftarrow W]\}$ . □

**Lemme 27.** *Pour tout lambda-terme fortement normalisable  $N$ , le w-terme  $\llbracket (N) \rrbracket_w$  est fortement normalisable.*

*Preuve.* Nous procédons par l'absurde. Supposons que  $M = \llbracket (N) \rrbracket_w$  a une suite de réduction infinie. Alors la stratégie perpétuelle sur  $M$  a un nombre infini de pas de  $\beta$ -réduction. De plus aucun pas de réduction n'est effectué dans un affaiblissement. Soit  $M_1, M_2, \dots$ , la stratégie perpétuelle sur  $M$ . Comme  $N = \lfloor M \rfloor$ , la suite  $\lfloor M_1 \rfloor, \lfloor M_2 \rfloor, \dots$ , est une suite de réduction infinie de  $N$ .  $\square$

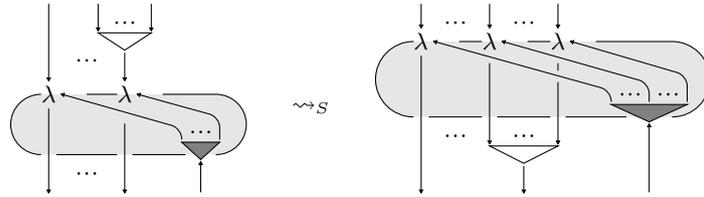
**Théorème 28.** *Le lambda-calcul atomique préserve la normalisation forte.*

*Preuve.* Pour tout lambda-terme  $N$ ,  $(N)$  est fortement normalisable si  $\llbracket (N) \rrbracket_w$  l'est, d'après le lemme 21, et  $\llbracket (N) \rrbracket_w$  est fortement normalisable si  $N$  l'est, par le lemme 27.  $\square$

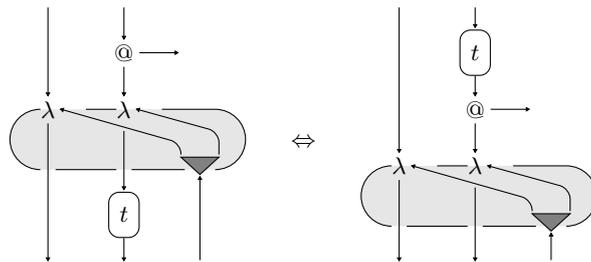
## 7. Conclusions et directions de recherche

Le lambda-calcul atomique présenté dans cet article est, dans la limite des connaissances des auteurs, le premier lambda-calcul qui possède la propriété d'atomicité, i.e. la possibilité de dupliquer et effacer les constructeurs de terme au fur et à mesure de la réduction. La principale nouveauté du calcul, le distributeur, peut certes apparaître au premier abord comme un peu alambiqué, mais il est conforté par une représentation graphique claire et possède une justification solide en théorie de la démonstration. Il permet d'introduire dans un cadre d'un calcul typé des techniques limitées jusqu'à maintenant à la réduction de graphe optimale.

Le présent article avait pour objectif de présenter les propriétés de base du lambda-calcul atomique, notamment la correspondance avec le lambda-calcul standard, la simulation de la  $\beta$ -réduction et la préservation de la normalisation forte. L'étape suivante sera d'étudier les propriétés fines de la réduction en lambda-calcul atomique et ses variantes. Une première modification, omise dans le présent article afin de préserver la simplicité de la présentation, consiste à permettre la réduction représentée graphiquement ci-dessous. Actuellement le distributeur bloque les partages qui le précèdent, par exemple dans le terme  $u[x_1, \dots, x_n \leftarrow y_i][y_1, \dots, y_m \leftarrow \lambda z.t^m]$ , jusqu'à ce que la duplication à l'intérieur du distributeur soit complètement effectuée. La réduction suivante corrige ce défaut d'une façon naturelle: elle clos le diagramme de confluence où deux noeuds de partage, l'un au-dessus de l'autre, rencontrent une abstraction.



Le calcul, dans son état actuel, a un défaut plus sérieux, mentionné dans la section 2: les abstractions implicites à l'intérieur d'un distributeur ne peuvent pas participer à un redex. Il y a cependant une voie prometteuse pour contourner le problème: quand une application agit sur un distributeur, il peut être possible de *permuter* des sous-termes de cette configuration de l'intérieur vers l'extérieur, ou vice versa, comme le suggère l'illustration suivante. Pour le lambda-calcul standard, des permutations de cette nature ont été étudiées dans [Reg94], [AK12]. Le lambda-calcul atomique est un cadre naturel pour les utiliser.



## Remerciements

Les auteurs tiennent à remercier les référés anonymes pour leurs commentaires bienveillants et fort avisés.

Ce travail a reçu le soutien du projet ANR–FWF Structural.

## Bibliographie

- [ACCL91] M. Abadi, Luca Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991.
- [AK12] Beniamino Accattoli and Delia Kesner. The permutative  $\lambda$ -calculus. In *LPAR*, pages 23–36, 2012.
- [Bal12] Thibaut Balabonski. A unified approach to fully lazy sharing. *POPL*, pages 469–480, 2012.
- [Bar84] Hendrik Pieter Barendregt. *The Lambda Calculus – Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.
- [BLM07] Tomasz Blanc, Jean-Jacques Lévy, and Luc Maranget. Sharing in the weak lambda-calculus revisited. In *Reflections on Type Theory, Lambda Calculus, and the Mind*, pages 41–50, 2007.
- [GGP10] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A proof calculus which reduces syntactic bureaucracy. In *LIPICs*, volume 6, pages 135–150, 2010.
- [Gug07] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.
- [Lam90] John Lamping. An algorithm for optimal lambda calculus reduction. In *POPL*, pages 16–30, 1990.
- [Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theor. Comput. Sci.*, 126(2):281–292, 1994.
- [Wad71] Christopher P. Wadsworth. *Semantics and Pragmatics of the Lambda Calculus*. PhD thesis, Oxford, 1971.