



HAL
open science

Etude polarisée du système L

Guillaume Munch-Maccagnoni

► **To cite this version:**

| Guillaume Munch-Maccagnoni. Etude polarisée du système L. 2008. inria-00295005v1

HAL Id: inria-00295005

<https://inria.hal.science/inria-00295005v1>

Preprint submitted on 11 Jul 2008 (v1), last revised 23 Jan 2009 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Étude polarisée du système L

Guillaume MUNCH–MACCAGNONI*

INRIA-Futurs Univ. of Pennsylvania

Résumé. Herbelin a proposé¹ le nom de *système L* pour référer à des syntaxes de termes propices à l'étude des calculs des séquents, dans lesquelles deux classes de *termes* interagissent au sein de *commandes*, à l'instar du calcul $\bar{\lambda}\mu\dot{\mu}$ de Curien et Herbelin ou du calcul dual de Wadler.

Le système L proposé ici dispose de constructeurs pour tous les connecteurs de la logique linéaire du second ordre, et troque l'interaction entre code et environnement pour un jeu entre *positifs* et *négatifs*. L fournit des quotients pour les calculs des séquents majeurs, dans leurs versions monolatères comme dans leurs versions bilatères, à savoir LL, LK et LLP.

Le lecteur logicielien appréciera le cadre unificateur pour l'étude calculatoire des séquents qu'il se propose d'être, tandis que le lecteur informaticien pourra apprécier le fait qu'il s'agit d'un pas dans la direction du projet d'Herbelin de refonder la théorie du calcul en traitant l'*appel par valeur* et l'*appel par nom* sur un pied d'égalité — en particulier L est impliqué vis-à-vis des stratégies de réduction, à savoir qu'un protocole d'élimination des coupures vérifiant la propriété de Church-Rosser pour L semble se distinguer, lequel permet de combiner des aspects paresseux et des aspects stricts.

L'outil principal pour l'étude de ce système est la *réalisabilité classique*, une conséquence étant que cet outil est désormais étendu à l'*appel par valeur*.

*Guillaume.Munch@ens-lyon.fr. Pré-publication, 10 juillet 2008.

¹[Her08]. La présence de trois diphtongues successives rendait sans doute $\bar{\lambda}\mu\dot{\mu}$ incommode à prononcer en anglais.

Table des matières

Introduction : la polarisation en informatique	1
1 Calcul dual : observations	3
2 Le système L	6
3 Expansion η et réversibilité des constructeurs négatifs	10
4 Relations d'élimination des coupures	12
5 Réalisabilité	14
6 La logique linéaire du second ordre	18
7 La logique classique du second ordre	22
8 Système L polarisé	23
9 La logique linéaire polarisée du second ordre	26
Conclusion	26

Introduction : la polarisation en informatique

La conception des stratégies de réduction qui prévaut dans les travaux sur la « *dualité du calcul* » : ceux de Curien et Herbelin [CH00, Her05, Her08] ou ceux de Wadler [Wad03], est que la stratégie de réduction ou bien donne aux valeurs le contrôle sur l'environnement, ou bien donne à l'environnement le contrôle sur les valeurs. Le premier cas permet d'implémenter l'*appel par valeur*, le second permet d'implémenter l'*appel par nom*.

Loin de s'y inscrire, ce travail a pour fil conducteur la thèse suivante :

Les notions informatiques de « paresse » et d'« avarice »² correspondent respectivement aux notions de « négatif » et de « positif » de la théorie de la démonstration.

Les correspondances négatif-paresseux et positif-avare est une chose l'on peut bien accepter dans l'ordre de l'intuitif. Il nous faudra cependant soutenir que « paresseux » et « avare » ne sont pas les qualités d'une stratégie d'évaluation, mais

²« Laziness » et « eagerness ». On a l'habitude de traduire eager par « strict », mais, c'est ennuyeux, strict ne possède pas de substantif convaincant.

les qualités des connecteurs de la logique. Cette thèse est défendue en section 1.

Historique Il sera question ici de dualité du calcul, de polarités, et de stratégie de réduction en calcul des séquents. Rappelons les travaux qui nous ont précédé et inspiré :

En 1999, Selinger [Sel01] introduit les *catégories de contrôle* ainsi que leurs duales, les *catégories de co-contrôle*, et montrent qu'elles admettent chacune pour langage interne un calcul $\lambda\mu$, en appel par nom pour les premières et en appel par valeur pour les secondes. Il en déduit un résultat de dualité syntaxique entre ces deux modes d'évaluation, que l'on nommera la *dualité du calcul*.

En 2000, Curien et Herbelin fournissent une interprétation calculatoire du calcul des séquents LK, le calcul $\bar{\lambda}\mu\bar{\mu}$ [CH00]. La symétrie hypothèse-conclusion du calcul des séquents permet alors une formulation claire de la dualité du calcul.

En 2002, Laurent [Lau02] inclut sa logique linéaire polarisée (LLP) dans les catégories de Selinger : les formules négatives deviennent des objets des catégories de contrôles, et les formules positives des objets des catégories de co-contrôle. La dualité entre positifs et négatifs répond donc à celle entre appel par nom et appel par valeur. Il suggère alors l'étude de la dualité du calcul à travers la notion de polarisation en logique.

C'est essentiellement la voie empruntée ici.

Par la suite, des variantes du calcul $\bar{\lambda}\mu\bar{\mu}$ sont apparues.

En 2003, Wadler [Wad03] donne une formulation du calcul des séquents de Gentzen avec une syntaxe parfaitement symétrique : le *calcul dual*. La conjonction et la disjonction additives y côtoient la négation ; l'implication est désormais définie par ces connecteurs.

En 2005, Herbelin [Her05] étudie toutes sortes de connecteurs dans son calcul, avec entre autres la conjonction multiplicative et sa duale, la disjonction multiplicative, qui s'apparente à la disjonction classique du calcul $\lambda\mu$ des catégories de Selinger.

Il est facile de remarquer que l'on peut faire correspondre aux multiplicatifs d'Herbelin les connecteurs \wp (« *par* ») et \otimes (« *tenseur* »), et aux additifs de Wadler les connecteurs \oplus (« *plus* ») et

$\&$ (« *avec* »). Il faut cependant une observation simple (section 1) sur ces connecteurs pour suggérer d'une part qu'ils possèdent une dynamique distincte, et d'autre part que la division entre *valeurs* et *environnements* du calcul $\bar{\lambda}\mu\bar{\mu}$ ou du calcul dual est moins importante que l'interaction entre *positifs* et *négatifs*. La possibilité était là de formuler un quotient de LLP avec une syntaxe à la $\bar{\lambda}\mu\bar{\mu}$.

Contrairement au calcul de Curien-Herbelin, les classes de nos termes ne détermineront donc pas si l'on est un terme ou un environnement, mais si l'on est un positif ou un négatif. On a donc choisi les classes en fonction des polarités afin de mettre en valeur cette interaction. Le sens premier d'une telle pratique est d'*internaliser* la dualité du calcul, c'est-à-dire que le couple de connecteurs \otimes/\wp pourra indifféremment signifier, du point de vue calculatoire, la *paire stricte* ou la *disjonction³ paresseuse*, et le couple de connecteurs $\&/\oplus$ signifier la *paire paresseuse* ou la *disjonction stricte* (section 2). Comme cela élimine la redondance des constructeurs, cela a en outre la conséquence de simplifier la syntaxe des termes.

Un tel calcul ne prend de sens en informatique qu'à condition de lever cette ambivalence. Ainsi, on pourra *observer* les commandes du système L de façon appel par valeur ou de façon appel par nom. On montrera alors que la lecture positive de LLP correspond à un calcul strict, implémentant l'« appel par valeur », alors que la lecture négative de LLP correspond à un calcul paresseux, implémentant l'« appel par nom » (section 8.2).

Enfin, on s'inspire de la négation de Wadler pour donner aux exponentielles ! (« *bien-sûr* ») et ? (« *pourquoi-pas* ») un sens calculatoire. Puisque la déréliction (l'introduction du ?) transforme un positif en un négatif, ce constructeur a pour principal effet de retarder l'évaluation de son argument. C'est pourquoi on le nomme « *quote* » et l'écrit $\backslash(\cdot)$, par référence au constructeur du langage Lisp.

Ce système « *polarisé* » s'étend facilement à LL et à LK^{tq} en supprimant les contraintes de polarisation. C'est ce dernier, qui prendra le nom de système L, que l'on présente ici.

Puisqu'il est question ici de stratégies d'élimination des coupures en calcul des séquents, on

³Le mot *somme* prête à confusion, \wp étant un multiplicatif.

mentionne enfin le travail de Danos, Joinet et Schellinx [DJS95]. Ce travail étudie les façons de réduire les séquents de LK^{tq} , lequel considérerait déjà les quatre connecteurs à la fois, dans le but de définir un cadre unificateur pour les différentes formulations de la logique classique. Différentes *colorations* (t ou q) des formules de LK^{tq} déterminent alors différentes façons d'éliminer des coupures, définies par le *tq-protocole*.

Remarquons que le seul (mais bon) argument des auteurs pour choisir ces connecteurs en particulier est le fait que la logique linéaire suggère qu'il peut en être ainsi, qui peut le plus pouvant le moins ([DJS95], p.7). Notre lecture des connecteurs est un progrès : nous affirmons qu'il est utile de considérer les quatre connecteurs à la fois car ceux-ci restent différents au niveau de la procéduralité, malgré l'absence de contraintes sur les règles structurelles (section 7).

La polarité ainsi que des considérations sur la réversibilité des connecteurs amèneront à définir sur le système L une stratégie d'élimination des coupures qui est analogue à celle du système LK_{pol}^{η} . L'argument le plus fort en faveur de cette stratégie est qu'elle permet de retrouver les stratégies d'évaluation usuelles de l'informatique.

Réalisations

1. On clarifie le lien entre *stratégies de réduction* et *polarité* (section 1), lien qui a été suggéré par Laurent ([Lau02], section « Dualité du calcul »).
2. On propose une même syntaxe asymétrique simple (section 2) pour les calculs des séquents du second ordre LL (section 6), LK^{tq} (section 7) et LLP (section 9) complets.
3. On la munit d'une stratégie d'élimination des coupures qui a la propriété de Church-Rosser, qui est compatible avec les expansions η , et qui permet de combiner des aspects paresseux à des aspects stricts (section 4).
4. On définit un outil de réalisabilité basé sur la réalisabilité classique de Krivine [Kri04] (section 5). Les trois logiques susmentionnées bénéficient du lemme d'adéquation. La réalisabilité classique trouve dans le système L un cadre naturel pour s'exprimer.

5. On répond au problème posé par Wadler [Wad04], qui a esquissé pour la logique linéaire une syntaxe à base de motifs. On prétend avoir résolu la question de façon satisfaisante (section 2.4).
6. On clarifie le lien entre calculs des séquents droits et calculs des séquents bilatéraux. Le système L permet les deux modes sans changement de syntaxe, en fonction de l'importance que l'on accorde à l'orientation des coupures (section 2.2).
7. On clarifie le lien entre le second ordre à la Girard [Gir06] (avec une syntaxe pour la généralisation et l'extraction) et le second ordre à la Krivine [Kri90] (l'introduction et l'élimination du \forall ne sont pas marqués dans la syntaxe des termes, ce qui permet de travailler dans le λ -calcul pur), section 2.1.

Notations On s'est efforcé de ne jamais s'écarter de la terminologie reçue sans de très sérieuses raisons. Les notations utilisées ici sont principalement issues du calcul $\bar{\lambda}\bar{\mu}\bar{\nu}$ tel que présenté dans l'article de Curien et Herbelin [CH00], ou dans le mémoire d'habilitation de ce dernier [Her05]. Lorsque l'informatique disposait de notations pré-existantes pour des notions de logique développées ici, on a choisi de les conserver. Ainsi, lorsqu'on affirme que \top (« top »), le neutre du $\&$, est le type de la constante *toplevel* tp , le lecteur est prié de ne pas y voir un mauvais jeu de mots.

1 Calcul dual : observations

On rappelle ici le calcul dual de Wadler, ce qui sera l'occasion de faire quelques observations en défense de notre thèse. Par souci de cohérence interne, on reformule celui-ci avec les notations proches du système L présenté par la suite. En particulier, les classes séparent les positifs des négatifs, et non les termes des co-termes. Les lieurs μ et $\bar{\mu}$ ne dénoteront donc pas respectivement un terme de valeur et un terme d'environnement, mais un terme positif et un terme négatif.

1.1 Syntaxe

Les *variables positives* sont $\{x, y, z \dots\}$ et les *variables négatives* $\{\alpha, \beta, \gamma \dots\}$. Les termes sont mutuellement définis par les grammaires suivantes :

$$\begin{array}{lcl}
 v & ::= & v_+ \quad | \quad v_\ominus \\
 v_+ & ::= & x \quad | \quad \mu\alpha.c \\
 & & | \quad \iota_i(v) \quad | \quad \text{not}(v) \quad (\oplus, \neg) \\
 v_\ominus & ::= & \alpha \quad | \quad \tilde{\mu}x.c \\
 & & | \quad (v|v) \quad | \quad \text{not}(v) \quad (\&, \neg) \\
 c & ::= & \langle v_+ \parallel v_\ominus \rangle \quad | \quad \langle v_\ominus \parallel v_+ \rangle
 \end{array}$$

Cette syntaxe est obtenue en repliant le calcul afin d'éliminer la redondance des constructeurs. L'orientation des commandes permet alors de déterminer si un terme agit en tant que *valeur* ou en tant qu'*environnement*. Par exemple, $\langle (v|v') \rangle$ est lu comme le constructeur de la paire, tandis que $|(v|v')\rangle$ représente le destructeur de la disjonction. Le *jeton* not dénote enfin l'échange entre *valeur* et *environnement*.

Les réductions de base sont :

$$\langle v_+ \parallel \tilde{\mu}x.c \rangle \rightarrow c[v_+/x] \quad (1)$$

$$\langle v_\ominus \parallel \mu\alpha.c \rangle \rightarrow c[v_\ominus/\alpha] \quad (2)$$

$$\langle \tilde{\mu}x.c \parallel v_+ \rangle \rightarrow c[v_+/x] \quad (3)$$

$$\langle \mu\alpha.c \parallel v_\ominus \rangle \rightarrow c[v_\ominus/\alpha] \quad (4)$$

Ces réductions forment des paires critiques. Chez Wadler, privilégier les deux premières permet d'implémenter l'appel par nom, tandis que privilégier les deux dernières permet d'implémenter l'appel par valeur.

not échange valeurs et environnement :

$$\langle \text{not}(v) \parallel \text{not}(v') \rangle \rightarrow \langle v' \parallel v \rangle$$

$\&$ interagit avec \oplus :

$$\langle \iota_i(v) \parallel (v_1 | v_2) \rangle \rightarrow \langle v \parallel v_i \rangle$$

$$\langle (v_1 | v_2) \parallel \iota_i(v) \rangle \rightarrow \langle v_i \parallel v \rangle$$

ces règles étant restreintes suivant la stratégie considérée.

Enfin, quelques règles supplémentaires permettent de s'assurer que les constructeurs (respectivement les destructeurs) s'évaluent de manière forte en appel par valeur (resp. par nom).

1.2 Multiplicatifs

Il est intéressant de comparer le comportement des multiplicatifs de [Her05] à celui des additifs du calcul dual. On enrichit la syntaxe de la façon suivante :

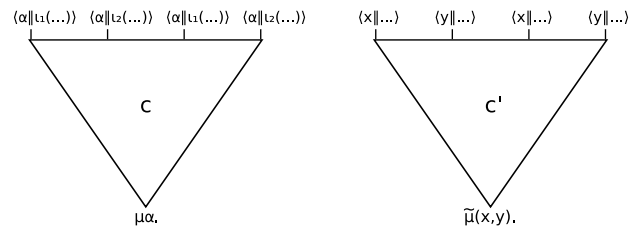
$$\begin{array}{lcl}
 v_+ & ::= & \dots \quad | \quad (v, v) \quad (\otimes) \\
 v_\ominus & ::= & \dots \quad | \quad \tilde{\mu}(x, y).c \quad (\wp)
 \end{array}$$

$\tilde{\mu}(x, y).c$ doit être lu comme un pattern matching sur le terme qui est amené à interagir avec lui (son *contre-terme*), (v, v') . Lors de la réduction, on a le choix entre les possibilités suivantes :

$$\begin{array}{l}
 \langle (v_1, v_2) \parallel \tilde{\mu}(x, y).c \rangle \rightarrow \langle v_1 \parallel \tilde{\mu}x.\langle v_2 \parallel \tilde{\mu}y.c \rangle \rangle \\
 \langle (v_1, v_2) \parallel \tilde{\mu}(x, y).c \rangle \rightarrow \langle v_2 \parallel \tilde{\mu}y.\langle v_1 \parallel \tilde{\mu}x.c \rangle \rangle \\
 \langle (v_1, v_2) \parallel \tilde{\mu}(x, y).c \rangle \rightarrow c[v_1/x, v_2/y] \quad (5)
 \end{array}$$

Le dilemme entre les deux premières met en évidence celui de l'ordre d'évaluation de la paire. La discussion suivante permet de rejeter la dernière.

On compare la géométrie de deux termes similaires, dont la seule différence est que l'un est formulé avec une disjonction additive \oplus et l'autre avec une disjonction multiplicative \wp , à savoir $\mu\alpha.c$ et $\tilde{\mu}(x, y).c'$:



On remarque tout d'abord que dans le cas du \oplus , les constructeurs sont aux feuilles, tandis que dans le cas du \wp , le constructeur est à la racine. La première conséquence est que le \oplus n'a besoin d'évaluer son contre-terme que lorsqu'il arrive à la racine. Il peut ainsi contrôler le flux d'exécution — il peut par exemple décider de *jeter* son contre-terme, lequel n'aura donc jamais la main à l'exécution. Cette faculté de pouvoir évaluer son contre-terme de manière paresseuse donne à \oplus un caractère strict.

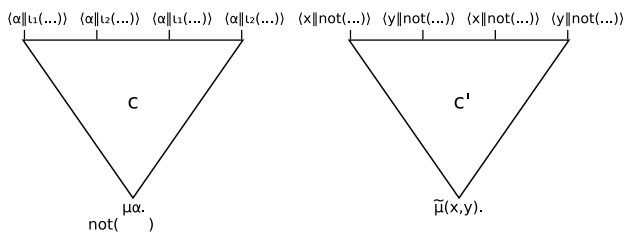
Le \wp quant à lui doit évaluer son contre-terme jusqu'à ce que celui-ci soit de la forme (v, v') , il a

donc un caractère paresseux, puisqu'il force \otimes à avoir un caractère strict. Une fois que son contre-terme est de la forme (v, v') seulement, survient le dilemme du choix de la règle de réduction. S'il s'agit de substituer v pour x et v' pour y , ainsi que le demande la règle de réduction (5) ci-dessus, alors le \wp finit par se comporter comme le \oplus : il peut choisir de jeter ses arguments sans les évaluer par exemple, ce qui lui donnerait un caractère strict. On rejette un tel comportement mi-chèvre mi-chou, c'est pourquoi on rejette la réduction (5).

La syntaxe est docile On remarque que le \oplus se comporte à l'inverse en \wp si l'on ne réduit le $\mu\alpha$ qu'une fois que le contre-terme est une valeur. En ajoutant des règles de réduction *ad hoc* pour réduire les sous-termes de $(v|v')$, on parvient à simuler le \wp . C'est la voie empruntée dans [Wad03] pour définir les réductions du calcul dual.

On verra en section 3 que ce qui achève de donner à \oplus son caractère strict est le fait de prendre au sérieux la réversibilité du $\&$.

Enter quote Le caractère *paresseux* ou *strict* est donc lié à la géométrie du constructeur : le $\tilde{\mu}(x, y)$ est *bloquant*⁴ alors que le $\mu\alpha$ ne l'est pas. En revanche ι_i l'est. On remarque qu'avec le not de wadler, on peut rendre $\mu\alpha$ bloquant comme ceci : $\text{not}(\mu\alpha\dots)$. On parvient ainsi à donner à \oplus et à \wp un comportement commun. Considérer :



que l'on fait interagir respectivement avec des programmes produisant du $\text{not}(v|v')$ et du $(\text{not}(v), \text{not}(v'))$. Les parties bloquantes sont situées aux mêmes endroits dans les deux termes, si bien que la réduction se déroule de façon similaire.

⁴Il est probable que l'on s'empressera d'inventer une règle qui permet de mettre tout terme de type \otimes sous la forme (\cdot, \cdot) afin me contredire. On ne reculera alors devant aucun travestissement : on aura simplement mimé la réversibilité du $\&$! La syntaxe est docile...

Cela donne, au niveau de la procéduralité, les équivalences de comportement suivantes :

$$\neg(A \oplus B) \simeq \neg A \wp \neg B$$

$$\neg(A^\perp \& B^\perp) \simeq \neg A^\perp \otimes \neg B^\perp$$

On interprète le not de wadler par la composée d'une exponentielle et de la négation linéaire⁵. Ces équations deviennent les équations bien connues :

$$?(A^\perp \& B^\perp) \simeq ?A^\perp \wp ?B^\perp$$

$$!(A \oplus B) \simeq !A \otimes !B$$

Ce qui met en évidence, du point de vue procédural, le caractère exponentiel du $\wp(\cdot)$ — hors toute règle de typage.

$\wp(\cdot)$ marquera ici l'introduction du pourquoi-pas (la déréluction). Il n'y a pas de progrès vis-à-vis des réseaux de preuve à faire cette présentation : l'exponentielle reste un point aveugle de la logique dans le sens où il s'agit ici d'un *token* dont on observe les effets mais pas la nature.

1.3 Conclusion

Ces observations font apparaître le calcul dual de Wadler comme un *jeu de mots*. Il s'agit d'utiliser la même notation pour dénoter tour à tour la paire paresseuse et la paire stricte (respectivement, la disjonction stricte et la disjonction paresseuse), à savoir le constructeur $(v|v')$ (resp. les destructeurs additifs ι_1 et ι_2). En effet, lorsque Wadler prétend résoudre la paire critique de façon appel par nom, il vient forcer la disjonction \oplus à se comporter en \wp , tandis que lorsqu'il prétend résoudre la paire critique de façon appel par valeur, il vient forcer la paire $\&$ à se comporter en \otimes .

Un indice à cela est le codage du tiers-exclus ou de l'implication employé dans [Wad03]. L'auteur donne :

$$\vdash \mu\alpha. \langle \iota_2(\text{not}(\tilde{\mu}x. \langle \iota_1(x) \parallel \alpha \rangle)) \parallel \alpha \rangle : A \oplus \neg A$$

pour preuve du tiers-exclus : pour l'appel par nom de Wadler, il évalue son contre-terme comme un tenseur : une fois ce dernier réduit

⁵Ce qui n'est pas tout neuf, cf. [Gir87], p.1 : « Abstract. The familiar connective of negation is broken into two operations: linear negation which is the purely negative part of negation and the modality "of course" [...] ».

2.1 Second ordre

Le λ -calcul admet deux traditions vis-à-vis de la généralisation (introduction du \forall) et de l'extraction (élimination du \forall) au second ordre : on peut les représenter dans la syntaxe de terme par les constructeurs $\lambda X.t$ et $\{t\}A$.⁶ D'autres auteurs⁷ ne les représentent pas : la syntaxe des termes est alors le λ -calcul pur. Cette différence n'est pas anodine : il ne s'agit pas simplement de choisir si l'on annote les termes par des types ou non (respectivement : le *style de Church* et le *style de Curry*). Dans le système F en appel par valeur, cela a une incidence sur la réduction des termes, puisque $\lambda X.t$ est une valeur.

On s'intéresse ici aux deux traditions :

1. Dans le cas où l'on choisit de ne pas refléter ces règles dans la syntaxe, cela signifie que \forall et \exists n'influent pas sur la polarité : il y aurait alors un $\forall X P$ positif et un $\exists X N$ négatif, bien que les connecteurs \forall et \exists sont traditionnellement, respectivement, négatif et positif.
2. S'il s'agit du contraire, à savoir importer l'équivalent du « \wedge » du système F dans le système L, le but est alors de prendre ces connecteurs, ainsi que leurs polarités, au sérieux : $\forall X A$ serait négatif et $\exists X A$ serait positif.

On verra par la suite que le projet d'un $\forall X P$ positif échoue (cf. remarque 5). En revanche, $\forall X N$ et $\exists X P$ ne posent aucun problème lorsqu'aucun constructeur ne leur correspond dans la syntaxe. Cela permettra l'adoption de la syntaxe « sans \wedge » pour les systèmes polarisés, comme LLP, dans lequel il n'y a pas de $\forall X P$.

Pour le moment, on utilisera donc la syntaxe $\{t\}$ pour l'extraction et $\tilde{\mu}\{\kappa\}.c$ pour la généralisation — ce sont les versions « non annotées » des syntaxes $A\{t\}$ et $\tilde{\mu}X\{\kappa\}.c$, où $\tilde{\mu}X\{\kappa\}.c$ est le pendant du λX du système F.

Cela revient en réalité, du point de vue calculatoire, à définir $\forall X P$ à partir de $\forall X N$ en posant $\forall X P \equiv \forall X ?P$ — nous revenons sur cette interprétation dans le système L polarisé, cf section 8.2.

⁶Notations d'après [Gir06].

⁷Cf. par exemple [Kri90, Kri93].

2.2 Séquents bilatéraux, séquents droits

La littérature admet deux traditions vis-à-vis des séquents : il y a d'une part les séquents bilatéraux, qui possèdent des formules des deux côtés du tourniquet \vdash , et les séquents droits, où toutes les formules sont à la droite de celui-ci. Ces derniers sont obtenus, lorsque les premiers évoluent dans un système symétriques, donc redondant, en définissant une négation involutive \perp sur les formules qui permet de replier les séquents bilatéraux $\Gamma \vdash \Delta$ en des séquents droits $\vdash \Gamma^\perp, \Delta$.

Notre syntaxe permet les deux approches :

1. On distingue les commandes $\langle v_+ \parallel v_\ominus \rangle$ et $\langle v_\ominus \parallel v_+ \rangle$ lorsque l'on veut distinguer entre *code* (ce qui est à la droite du tourniquet) et *environnement* (ce qui est à la gauche du tourniquet), en imposant l'ordre de la commande de la façon suivante, en héritage du calcul $\bar{\lambda}\mu\tilde{\mu}$:

$$\frac{\Gamma \vdash v : A; \Delta \quad \Gamma'; v' : A \vdash \Delta'}{\langle v \parallel v' \rangle : (\Gamma, \Gamma' \vdash \Delta, \Delta')} \text{ (cut)}$$

Donc le code est à gauche dans la commande, et l'environnement à droite, à la façon des machines abstraites, comme la machine de Krivine [Kri04].

Cela permet de faire de L un calcul mélangeant des aspects *paresseux* et des aspects *stricts* . Par exemple, dans une commande $\langle v_+ \parallel v_\ominus \rangle$, le couple \otimes/\wp implémente la paire stricte et le couple $\oplus/\&$ la somme stricte, tandis que dans une commande $\langle v_\ominus \parallel v_+ \rangle$, le couple \otimes/\wp implémente la somme paresseuse et le couple $\oplus/\&$ la paire paresseuse.

2. On pose $\langle v_+ \parallel v_\ominus \rangle = \langle v_\ominus \parallel v_+ \rangle$ pour tous v_+ et v_\ominus dans le cas des séquents droits, et les commandes sont alors de façon non ambiguë typées par :

$$\frac{\vdash v : A; \Gamma \quad \vdash v' : A^\perp; \Delta}{\langle v \parallel v' \rangle : (\vdash \Gamma, \Delta)} \text{ (cut)}$$

Dans la suite, on s'intéresse aux séquents droits, c'est pourquoi l'on pose à partir de maintenant, pour tous v_+ et v_\ominus , $\langle v_+ \parallel v_\ominus \rangle = \langle v_\ominus \parallel v_+ \rangle$.

2.3 Un peu de terminologie

Il est important de clarifier les mots que l'on emploie car certaines homonymies bien pratiques pourraient autrement prêter à confusion. On distingue les *connecteurs* (logiques) des *constructeurs* (dans la syntaxe). Ainsi, les *constructeurs* \otimes , \wp , $\&$ et \oplus sont respectivement $\langle \cdot, \cdot \rangle$, $\tilde{\mu}(\cdot, \cdot)$, $\tilde{\mu}(\iota_1(\cdot) \cdot | \iota_2(\cdot) \cdot)$ et $\iota_i(\cdot)$. Les *connecteurs* \otimes , \wp , $\&$ et \oplus réfèrent respectivement à la *paire stricte*, la *disjonction paresseuse*, la *paire paresseuse* et la *disjonction stricte*.

Constructeurs et connecteurs sont reliés de la façon suivante, lorsqu'on distingue code et environnement :

- Constructeur \otimes : $\langle \cdot, \cdot \rangle$ est le constructeur de la paire stricte et $|(\cdot, \cdot)$ est le *destructeur* de la disjonction paresseuse.
- Constructeur \wp : $\langle \tilde{\mu}(\cdot, \cdot) \cdot |$ est le constructeur de la disjonction paresseuse et $|\tilde{\mu}(\cdot, \cdot) \cdot \rangle$ est le destructeur de la paire stricte.
- Constructeur $\&$: $\langle \tilde{\mu}(\iota_1(\cdot) \cdot | \iota_2(\cdot) \cdot) \rangle$ (ou encore : $\langle (\cdot | \cdot) \rangle$) est le constructeur de la paire paresseuse et $|\tilde{\mu}(\iota_1(\cdot) \cdot | \iota_2(\cdot) \cdot) \rangle$ est le destructeur de la disjonction stricte.
- Constructeur \oplus : $\langle \iota_i(\cdot) \rangle$ est le constructeur de la somme stricte et $|\iota_i(\cdot) \rangle$ est le destructeur de la paire paresseuse.

Le constructeur \otimes (resp. \oplus) est le dual du constructeur \wp (resp. $\&$) pour l'involution \cdot^\perp . Le connecteur \otimes (resp. \oplus) est le dual du connecteur \wp (resp. $\&$) pour la dualité du calcul.

2.4 Motifs

Notation 1. On se permet de composer les motifs du μ et du $\tilde{\mu}$. Cela permet une concision attrayante. Sans rentrer dans une définition exhaustive, on pourra utiliser par exemple la notation :

$$\tilde{\mu}(\iota_1(x, y) \cdot c_1 | \iota_2(z) \cdot c_2)$$

pour remplacer :

$$\tilde{\mu}(\iota_1(a) \cdot \langle a | \tilde{\mu}(x, y) \cdot c_1 \rangle | \iota_2(z) \cdot c_2)$$

lorsque $a \notin \mathcal{FV}(c_1)$. De même, on pourra utiliser la notation :

$$\mu^{\setminus}(x, y) \cdot \langle y | \setminus(v) \rangle$$

au lieu de :

$$\mu^{\setminus}(z) \cdot \langle z | \tilde{\mu}(x, y) \cdot \langle y | \setminus(v) \rangle \rangle$$

lorsque $z \notin \mathcal{FV}(v)$.

Avoir des motifs pour citoyens de première classe est une piste intéressante. Cela permettrait de formuler les règles de la logique en distinguant la création du motif de son activation (ou liaison), par exemple pour le \wp , pour lequel la création du motifs (x, y) dans l'environnement à partir des motifs x et y serait disjointe de l'activation $\tilde{\mu}$:

$$\frac{\vdash v : A; x : P, y : Q, \Gamma}{\vdash v : A; (x, y) : P \wp Q; \Gamma} (\wp)$$

On remarque que le séquent ne garde pas de trace de l'endroit où le motif a été formé : cela correspond donc à un quotient qui prend au sérieux la réversibilité du \wp .

Cela permettrait également de rendre les règles η des différents constructeurs instances des règles η originelles, *i.e.*

$$\mu x. \langle x | \alpha \rangle =_{\eta} \alpha$$

serait étendu au cas où x serait un motif. Cela permettrait aussi d'identifier par exemple :

$$\tilde{\mu} x. \langle (x, x) | \alpha \rangle$$

et :

$$\tilde{\mu}(y, z) \cdot \langle ((y, z), (y, z)) | \alpha \rangle$$

modulo α -conversion, ce qui exprime également la réversibilité du \wp .

Nous ne sommes pas les premiers à remarquer que le calcul des séquents correspond à un calcul de motifs (cf. par exemple les travaux de Kesner, [BTKP93]). Par ailleurs, dans [Wad04], Wadler propose un calcul de purs motifs pour la logique linéaire.

Malheureusement, ce travail laisse aussi préfigurer la bureaucratie qu'entraînerait une telle *simplification* du système L : cela devient peu tractable lorsque pour les additifs, il nous faut gérer une *forêt* de termes.

D'autre part, penser par exemple que pour effectuer la contraction :

$$\vdash \Gamma, x : A, y : A \Rightarrow \vdash \Gamma, x \cup y : A$$

il est nécessaire d'*unifier* les motifs x et y en le motif $x \cup y$ pour en extraire les substitutions de variables à appliquer au terme, et qu'il est donc nécessaire de s'assurer qu'une telle unification réussit. En particulier, l'*occurs-check* doit réussir — ce qui est facilement rempli si les règles axiome sont de la forme

$$\vdash i : A; i : A^\perp$$

avec i une variable atomique. Imaginer cependant la lourdeur bureaucratique des nouveaux *lemmes syntaxiques* à prouver !

Pour cette raison, on ne pense pas opportun de développer un tel calcul avec des motifs de première classe. C'est que, les avantages que cela procurerait, on peut très bien les obtenir de façon informelle.

En effet, d'une part, on prétend qu'en faisant de la construction de motifs une convention de notation comme dans la notation 1, et en introduisant la réversibilité des constructeurs négatifs explicitement dans les règles de réduction (cf. la relation \rightarrow_r plus bas), on obtient le même quotient des calculs des séquents que si les motifs étaient de première classe.

D'autre part, on peut permettre aux environnements de contenir des motifs : il suffit de considérer cela comme une astuce informelle de dérivation.

Ce que l'on propose en réponse à [Wad04] est un *informalisme* : cela se fait donc à bureaucratie constante.

2.5 Valeurs et commandes sans coupures

Définition 2 (Valeurs). On définit les classes des *valeurs* et des *valeurs positives*, cas particulier des termes et des termes positifs, par la grammaire :

$$\begin{aligned} V &= V_+ \mid v_\ominus \\ V_+ &= x \mid (V, V) \mid \iota_i(V) \mid \mu^i(\kappa).c \mid \{V\} \mid () \end{aligned}$$

On note \mathbb{V} l'ensemble des valeurs.

On définit ensuite la classe des *valeurs négatives* :

$$\begin{aligned} V_\ominus &= \alpha \mid \tilde{\mu}(\kappa, \kappa').c \mid \tilde{\mu}(\iota_1(\kappa).c \mid \iota_2(\kappa').c') \\ &\quad \mid \backslash(V) \mid \tilde{\mu}\{\kappa\}.c \mid \tilde{\mu}().c \mid \text{tp} \end{aligned}$$

Remarquer que $\backslash(v)$ et $\tilde{\mu}x.c$ sont des valeurs sans être des valeurs négatives.

Définition 3. Une commande est une *coupure* si elle n'est d'aucune des formes suivantes :

$$\begin{aligned} \langle V_+ \parallel \alpha \rangle \\ \langle V_+ \parallel \text{tp} \rangle \\ \langle x \parallel V_\ominus \rangle \end{aligned}$$

Une commande est *sans coupure* si aucune de ses sous-commandes n'est une coupure, *i.e.* si toutes ses sous-commandes sont de l'une des formes ci-dessus.

2.6 μ - et $\tilde{\mu}$ -réductions

Définition 4. On définit trois relations sur \mathcal{C}^0 :

- $\langle \mu\alpha.c \parallel v_\ominus \rangle \rightarrow_\mu c[v_\ominus/\alpha]$ pour tous v_\ominus, c .
- $\langle V_+ \parallel \tilde{\mu}x.c \rangle \rightarrow_{\tilde{\mu}} c[V_+/x]$ pour tous V_+, c .
- $\langle v_+ \parallel \tilde{\mu}x.c \rangle \rightarrow_{\tilde{\mu}^v} c[v_+/x]$ pour tous v_+, c .

Notation 2. Par commodité, $\tilde{\mu}\kappa$ dénote $\mu\alpha$ si κ est de la forme α , ou $\tilde{\mu}x$ si κ est de la forme x .

Remarque 1. On peut reformuler $\rightarrow_\mu \cup \rightarrow_{\tilde{\mu}}$ par :

$$\langle \tilde{\mu}\kappa.c \parallel V \rangle (\rightarrow_\mu \cup \rightarrow_{\tilde{\mu}}) c[V/\kappa]$$

2.7 β -réductions

Définition 5. On définit une relation sur \mathcal{C} , que l'on note \rightarrow_β (fig. 1).

Remarque 2. Pour l'élimination des coupures \otimes/\wp , on a fait un choix arbitraire concernant les profondeurs relatives de v et de v' . Par exemple on a préféré :

$$\langle (v_+, v'_+) \parallel \tilde{\mu}(x, y).c \rangle \rightarrow_\beta \langle v_+ \parallel \tilde{\mu}x.\langle v'_+ \parallel \tilde{\mu}y.c \rangle \rangle$$

à :

$$\langle (v_+, v'_+) \parallel \tilde{\mu}(x, y).c \rangle \rightarrow_\beta \langle v'_+ \parallel \tilde{\mu}y.\langle v_+ \parallel \tilde{\mu}x.c \rangle \rangle$$

ce qui correspond à donner priorité au côté gauche dans la paire (v_+, v'_+) .

Le comportement asymétrique de notre connecteur \otimes est à relier aux foncteurs binôïdaux à la base des catégories de contrôle et de co-contrôle [Sel01]. Plus précisément, le fait que (v_1, v_2) n'est pas calculatoirement isomorphe à (v_2, v_1) — ou encore : la distinction entre l'évaluation de gauche à droite de celle

$$\begin{array}{lcl}
\langle (v, v') \parallel \tilde{\mu}(\kappa, \kappa').c \rangle & \rightarrow_{\beta} & \langle v \parallel \tilde{\mu}\kappa.\langle v' \parallel \tilde{\mu}\kappa'.c \rangle \rangle \quad (\otimes/\wp) \\
\langle \iota_i(v) \parallel \tilde{\mu}(\iota_1(\kappa_1).c_1 \mid \iota_2(\kappa_2).c_2) \rangle & \rightarrow_{\beta} & \langle v \parallel \tilde{\mu}\kappa_i.c_i \rangle \quad (\oplus_i/\&) \\
\langle () \parallel \tilde{\mu}().c \rangle & \rightarrow_{\beta} & c \quad (\mathbf{1}/\perp) \\
\langle \mu^{\lambda}(\kappa).c \parallel ^{\lambda}(v) \rangle & \rightarrow_{\beta} & \langle v \parallel \tilde{\mu}\kappa.c \rangle \quad (!/?) \\
\langle \{v\} \parallel \tilde{\mu}\{\kappa\}.c \rangle & \rightarrow_{\beta} & \langle v \parallel \tilde{\mu}\kappa.c \rangle \quad (\exists/\forall)
\end{array}$$

Dans le cas où les polarités des variables sont telles que le membre de droite n'est pas défini, la relation \rightarrow_{β} n'est pas définie.

FIG. 1: La réduction \rightarrow_{β} .

de droite à gauche — correspond au fait que $A \otimes B \xrightarrow{(f \otimes B); (A' \otimes g)} A' \otimes B'$ est en général différent de $A \otimes B \xrightarrow{(A \otimes g); (f \otimes B')} A' \otimes B'$ dans les catégories de co-contrôle. Définir $f \otimes g$ en forçant l'égalité des deux précédents entraîne l'effondrement de ces catégories.

3 Expansion η et réversibilité des constructeurs négatifs

Les trois relations suivantes expriment la réversion des constructeurs \wp , $\&$, \forall et \perp :

$$\begin{array}{l}
v_{\ominus} \rightarrow_{\eta} \tilde{\mu}(\kappa, \kappa').\langle (\kappa, \kappa') \parallel v_{\ominus} \rangle \\
v_{\ominus} \rightarrow_{\eta} \tilde{\mu}(\iota_1(\kappa).\langle \iota_1(\kappa) \parallel v_{\ominus} \rangle \mid \iota_2(\kappa).\langle \iota_2(\kappa) \parallel v_{\ominus} \rangle) \\
v_{\ominus} \rightarrow_{\eta} \tilde{\mu}\{\kappa\}.\langle \{\kappa\} \parallel v_{\ominus} \rangle \\
v_{\ominus} \rightarrow_{\eta} \tilde{\mu}().\langle () \parallel v_{\ominus} \rangle
\end{array}$$

Elles expriment également l'expansion η de ces constructeurs. En effet, les η -expansions des règles axiome :

$$\begin{array}{l}
\vdash \alpha : P \wp Q; \alpha : P^{\perp} \otimes Q^{\perp} \\
\vdash \alpha : P \& Q; \alpha : P^{\perp} \oplus Q^{\perp}
\end{array}$$

sont respectivement :

$$\vdash \tilde{\mu}(x, y).\langle (x, y) \parallel \alpha \rangle : P \wp Q; \alpha : P^{\perp} \otimes Q^{\perp}$$

et :

$$\vdash \tilde{\mu}(\iota_1(x).\langle \iota_1(x) \parallel \alpha \rangle \mid \iota_2(y).\langle \iota_2(y) \parallel \alpha \rangle) : P \& Q; \alpha : P^{\perp} \oplus Q^{\perp}$$

Coups respectivement avec $\vdash v_{\ominus} : P \wp Q; \Gamma$ et $\vdash v_{\ominus} : P \& Q; \Gamma$, cela donne *après μ -réduction* les mêmes termes que ceux que l'on obtient lorsque l'on prouve la réversibilité des constructeurs \wp et

$\&$, à savoir :

$$\tilde{\mu}(\kappa, \kappa').\langle (\kappa, \kappa') \parallel v_{\ominus} \rangle$$

et :

$$\tilde{\mu}(\iota_1(\kappa).\langle \iota_1(\kappa) \parallel v_{\ominus} \rangle \mid \iota_2(\kappa).\langle \iota_2(\kappa) \parallel v_{\ominus} \rangle)$$

(Et de même pour \forall et \perp .)

Permettre ces transformations implique l'évaluation forte, respectivement, du \otimes , du \oplus , du \exists et du $?$. En effet, on observe que si l'on veut réduire par exemple $\langle (v_+, v'_+) \parallel v_{\ominus} \rangle$, on obtient par \rightarrow_{η} :

$$\langle (v_+, v'_+) \parallel \tilde{\mu}(x, y).\langle (x, y) \parallel v_{\ominus} \rangle \rangle$$

ce qui par \rightarrow_{β} entraîne l'évaluation de v_+ puis éventuellement celle de v'_+ .

C'est vrai en particulier si v_{\ominus} est une variable. Elles permettent donc aussi de « décomposer » les variables négatives, dans le sens où la présence d'une variable dans l'un des deux membres n'empêche pas la poursuite du calcul.

3.1 Une η -expansion pour $! ?$

On considère l' η -expansion de ! :

$$v_+ \rightarrow_{\eta, v} \mu^{\lambda}(\kappa).\langle v_+ \parallel ^{\lambda}(\kappa) \rangle$$

On considère également la réversibilité de ! correspondante : si v_+ n'est pas de la forme $\mu^{\lambda}(x).c$ et si v n'est pas une valeur, alors on pose :

$$\langle v_+ \parallel ^{\lambda}(v) \rangle \rightarrow_{r, v} \langle \mu^{\lambda}(\kappa).\langle v_+ \parallel ^{\lambda}(\kappa) \rangle \parallel ^{\lambda}(v) \rangle$$

On peut légitimement se demander s'il ne faut

pas également accepter l'expansion $\eta_{!v}$. Le raisonnement précédent s'applique-t-il ici? La réponse est intéressante.

La forme η -expansée de la règle axiome $\vdash y : !N; y : ?N^\perp$ est bien :

$$\vdash \mu^{\setminus}(x). \langle y \parallel^{\setminus}(x) \rangle : !N; y : ?N^\perp$$

Puis, si l'on coupe avec $\vdash v : !N; \Gamma$, le terme que l'on obtient ne se réduit pas par μ -réduction mais par $\tilde{\mu}$ -réduction. Si l'on accepte cette réduction, on obtient :

$$\vdash \mu^{\setminus}(x). \langle v_+ \parallel^{\setminus}(x) \rangle : !N; \Gamma$$

ce qui ressemble à une réversibilité de !.

Cependant, la promotion impose une contrainte sur le contexte dans les systèmes usuels, ce qui rend ce dernier terme non-dérivable si la contrainte n'est pas remplie. Trois cas se présentent :

LL : Γ doit être de la forme $? \Gamma'$ pour LL, ce qui n'est pas vérifié en général; LL ne valide donc pas la réversibilité de !.

LLP : Γ doit être composé de négatifs uniquement pour LLP. Dans $\vdash_{LLP} v : !N; \Gamma$, la formule positive est $!N$, donc les formules de Γ sont négatives et LLP valide donc la réversibilité de !.

LK : Il n'y a pas de contrainte sur Γ dans le cas de LK, qui valide donc la réversibilité de !.

On observe que la réversibilité de ! a pour conséquence majeure de forcer l'évaluation de v dans $\setminus(v)$. Donc, si l'on ne dispose pas de la réversibilité de !, il est alors nécessaire que $\setminus(v)$ soit une valeur — l'exécution de v est « retardée ». D'ailleurs, les catégories non dégénérées pour la logique classique, à savoir les catégories de (co-)contrôle de Selinger [Sel01, Sel03] ainsi que les \mathcal{C} -catégories de Führmann et Thielecke [FT04] ajoutent aux catégories cartésiennes fermées des constructions introduisant la notion d'ordre d'évaluation :

- Dans les catégories de co-contrôle, le morphisme $(f \otimes B); (A' \otimes g)$ est généralement distinct de $(A \otimes f); (g \otimes B')$, ce qui correspond à donner de l'importance à l'ordre d'évaluation de la paire ;

- Les \mathcal{C} -catégories sont basées sur un foncteur L où Lf est f dont on « retarde » l'évaluation.

Similairement, on affirme que $\setminus(v)$ doit « retarder » l'évaluation de v , et donc que $\rightarrow_{r_!v}$ ne doit pas faire partie de la stratégie d'évaluation. En effet, si l'on suppose qu'on permet $\rightarrow_{r_!v}$, et en même temps l'affaiblissement sur les positifs, qui est une hypothèse réaliste, alors on a à la fois :

$$\langle \mu_-.c \parallel^{\setminus}(\mu_-.c') \rangle \rightarrow_{\mu} c$$

et :

$$\begin{aligned} \langle \mu_-.c \parallel^{\setminus}(\mu_-.c') \rangle &\rightarrow_{r_!v} \langle \mu^{\setminus}(x). \langle \mu_-.c \parallel^{\setminus}(x) \rangle \parallel^{\setminus}(\mu_-.c') \rangle \\ &\rightarrow_{\beta} \langle \mu_-.c' \parallel^{\setminus} \tilde{\mu}x. \langle \mu_-.c \parallel^{\setminus}(x) \rangle \rangle \\ &\rightarrow_{\mu} c' \end{aligned}$$

sans que l'on ne dispose d'un argument du type polarité pour résoudre cette paire critique, puisqu'elle se fait entre deux *positifs* $\mu_-.c$ et $\mu_-.c'$.

Comme LLP force la linéarité sur les positifs, permettre à $\setminus(v)$ d'être évalué de manière forte n'entraînerait pas la dégénérescence du calcul. LK, quant à lui, est dégénéré.

On rejette donc la transformation $\eta_!$ d'après l'intuition que $\setminus(v)$ doit être une valeur. On remarque que ce qui est fautif, ce n'est pas l' η -expansion de la règle axiome pour $!A$ puisqu'on la dérive aisément dans LL comme dans LLP.

En revanche, il nous faut rejeter la $\tilde{\mu}$ -réduction sur les non-valeurs au moins dans ce cas, puisqu'elle implique $\eta_{!v}$. Autrement dit, la $\tilde{\mu}$ -réduction sur les non-valeurs ne préserve pas la prouvabilité, dans le cas général, dans les « bons » systèmes LL et LLP.

On note cependant que la réversion de ! est validée dans le cas des valeurs, et qu'on peut donc la considérer dans une stratégie d'évaluation. En effet, on peut admettre :

$$V_+ \rightarrow_{\eta} \mu^{\setminus}(\kappa). \langle V_+ \parallel^{\setminus}(\kappa) \rangle$$

car, que V_+ soit une variable ou une promotion, son contexte est de la forme adéquate, par exemple $? \Gamma$ dans le cas de LL. Cela permet par exemple de réduire $\langle x \parallel^{\setminus}(\mu\alpha.c) \rangle$.

3.2 Équivalence η

Définition 6. La relation \rightarrow_{η} est définie sur $\mathcal{V}_+ \cup \mathcal{V}_\ominus$ en fig. 2.

$v_+ \rightarrow_{\eta} \mu\alpha.\langle v_+ \parallel \alpha \rangle$	(η_+)
$v_{\ominus} \rightarrow_{\eta} \tilde{\mu}x.\langle x \parallel v_{\ominus} \rangle$	(η_{\ominus})
$v_{\ominus} \rightarrow_{\eta} \tilde{\mu}(\kappa, \kappa').\langle (\kappa, \kappa') \parallel v_{\ominus} \rangle$	(η_{\wp})
$v_{\ominus} \rightarrow_{\eta} \tilde{\mu} \left(\iota_1(\kappa).\langle \iota_1(\kappa) \parallel v_{\ominus} \rangle \mid \iota_2(\kappa).\langle \iota_2(\kappa) \parallel v_{\ominus} \rangle \right)$	$(\eta_{\&})$
$v_{\ominus} \rightarrow_{\eta} \tilde{\mu}\{\kappa\}.\langle \{\kappa\} \parallel v_{\ominus} \rangle$	(η_{\forall})
$v_{\ominus} \rightarrow_{\eta} \tilde{\mu}().\langle () \parallel v_{\ominus} \rangle$	(η_{\perp})
$V_+ \rightarrow_{\eta} \mu^{\lambda}(\kappa).\langle V_+ \parallel ^{\lambda}(\kappa) \rangle$	$(\eta_!)$

Pour des variables κ, κ' faisant concorder les polarités.

FIG. 2: La réduction \rightarrow_{η} sur les termes.

On étend cette relation aux commandes. Cette nouvelle relation n'est pas bonne pour définir un protocole d'élimination des coupures, mais traduit une équivalence observationnelle.

Définition 7. La relation \rightarrow_{η} est définie sur \mathcal{C} en fig. 3.

3.3 Relation de réversion

La relation de réversion \rightarrow_r est la relation \rightarrow_{η} que l'on restreint de façon à ce qu'elle puisse définir un protocole d'élimination des coupures :

Définition 8. La relation \rightarrow_r est définie sur \mathcal{C} par, pour des variables κ, κ' faisant concorder les polarités :

- Si (v, v') n'est pas une valeur, et que l'on a $\langle (v, v') \parallel v_{\ominus} \rangle \not\rightarrow_{\beta}$, alors on pose :

$$\langle (v, v') \parallel v_{\ominus} \rangle \rightarrow_r \left\langle (v, v') \parallel \tilde{\mu}(\kappa, \kappa').\langle (\kappa, \kappa') \parallel v_{\ominus} \rangle \right\rangle$$

- Si $\iota_i(v)$ n'est pas une valeur, et que l'on a $\langle \iota_i(v) \parallel v_{\ominus} \rangle \not\rightarrow_{\beta}$, alors on pose :

$$\langle \iota_i(v) \parallel v_{\ominus} \rangle \rightarrow_r \left\langle \iota_i(v) \parallel \tilde{\mu} \left(\iota_1(\kappa).\langle \iota_1(\kappa) \parallel v_{\ominus} \rangle \mid \iota_2(\kappa).\langle \iota_2(\kappa) \parallel v_{\ominus} \rangle \right) \right\rangle$$

- Si $\{v\}$ n'est pas une valeur, et que l'on a $\langle \{v\} \parallel v_{\ominus} \rangle \not\rightarrow_{\beta}$, alors on pose :

$$\langle \{v\} \parallel v_{\ominus} \rangle \rightarrow_r \left\langle \{v\} \parallel \tilde{\mu}\{\kappa\}.\langle \{\kappa\} \parallel v_{\ominus} \rangle \right\rangle$$

- Si $^{\lambda}(v)$ n'est pas une valeur négative, et que l'on a $\langle V_+ \parallel ^{\lambda}(v) \rangle \not\rightarrow_{\beta}$, alors on pose :

$$\langle V_+ \parallel ^{\lambda}(v) \rangle \rightarrow_r \left\langle \mu^{\lambda}(\kappa).\langle V_+ \parallel ^{\lambda}(\kappa) \rangle \parallel ^{\lambda}(v) \right\rangle$$

4 Relations d'élimination des coupures

4.1 L'élimination de tête

Définition 9 (Élimination de tête). On définit la relation \rightarrow_h (h pour « head ») sur \mathcal{C} en posant :

$$\rightarrow_h = \rightarrow_{\mu} \cup \rightarrow_{\tilde{\mu}} \cup \rightarrow_{\beta} \cup \rightarrow_r$$

Définition 10 (Termes mal-formés). Une commande c qui est une coupure est dite *malformée* dès qu'elle n'est pas un redex pour \rightarrow_h .

Proposition 1. Pour toute commande c , on a $c \not\rightarrow_h$ si et seulement si : c est une coupure implique c est malformée.

Cela découle de la définition d'être mal-formé — le contenu de cette proposition est donc à chercher dans la caractérisation suivante des termes mal formés :

Proposition 2 (Termes mal-formés, caractérisation). On désigne par $v_{\otimes}, v_{\oplus}, v_{\exists}, v_1, v_!, v_{\wp}, v_{\&}, v_{\forall}, v_{\perp}$, et $v_?$ des termes respectivement de la forme (v, v') , $\iota_i(v)$, $\{v\}$, $()$, $\mu^{\lambda}(\kappa).c$, $\tilde{\mu}(\kappa, \kappa').c$, $\tilde{\mu}(\iota_1(\kappa).c \mid \iota_2(\kappa').c')$, $\tilde{\mu}\{x\}.c$, $\tilde{\mu}().c$ et $^{\lambda}(v)$. Une commande malformée est de la forme $\langle v_{\otimes} \parallel v_{\oplus'} \rangle$ avec \otimes parmi $\otimes, \oplus, \exists, !$ et \oplus' parmi $\wp, \&, \forall, \perp, ?$, avec en plus l'une des deux propriétés équivalentes suivantes :

$$\begin{aligned}
\langle (v, v') \parallel v_{\ominus} \rangle &\rightarrow_{\eta} \langle (v, v') \parallel \tilde{\mu}(\kappa, \kappa'). \langle (\kappa, \kappa') \parallel v_{\ominus} \rangle \rangle & (\eta_{\&}) \\
\langle \iota_i(v) \parallel v_{\ominus} \rangle &\rightarrow_{\eta} \langle \iota_i(v) \parallel \tilde{\mu}(\iota_1(\kappa). \langle \iota_1(\kappa) \parallel v_{\ominus} \rangle \mid \iota_2(\kappa). \langle \iota_2(\kappa) \parallel v_{\ominus} \rangle) \rangle & (\eta_{\&}) \\
\langle \{v\} \parallel v_{\ominus} \rangle &\rightarrow_{\eta} \langle \{v\} \parallel \tilde{\mu}\{\kappa\}. \langle \{\kappa\} \parallel v_{\ominus} \rangle \rangle & (\eta_{\vee}) \\
\langle () \parallel v_{\ominus} \rangle &\rightarrow_{\eta} \langle () \parallel \tilde{\mu}(). \langle () \parallel v_{\ominus} \rangle \rangle & (\eta_{\perp}) \\
\langle V_+ \parallel \cdot (v) \rangle &\rightarrow_{\eta} \langle \mu^{\cdot}(\kappa). \langle V_+ \parallel \cdot (\kappa) \rangle \parallel \cdot (v) \rangle & (\eta_{!})
\end{aligned}$$

Pour des variables κ, κ' faisant concorder les polarités.

FIG. 3: La réduction \rightarrow_{η} sur les commandes.

1. Elle n'est un redex ni pour \rightarrow_{β} , ni pour \rightarrow_{τ} .
2. v_{\otimes} est une valeur et $v_{\otimes'}$ est une valeur négative, avec l'une des suivantes :
 - a) $\otimes' \neq \otimes^{\perp}$;
 - b) Les polarités des sous-termes du v_{\otimes} , du v_{\oplus} , du v_{\exists} ou du $v_{?}$ ne correspondent pas à celles des variables liées par le $v_{\&}$, le $v_{\&}$, le v_{\vee} ou le $v_{!}$.

Remarque 3. La commande suivante n'est donc pas malformée :

$$\langle (\mu_{-}.c, \text{toto}) \parallel \tilde{\mu}(\alpha, x).c' \rangle$$

Elle se réduit d'ailleurs en c . Pourtant, la variable α est de la mauvaise polarité et toto est un terme fantaisiste. C'est un comportement que l'on doit avoir si l'on tient à la proposition 4.

Proposition 3. \rightarrow_h n'admet pas de paire critique, i.e. si $c \rightarrow_h c'$ et $c \rightarrow_h c''$, alors $c' = c''$.

Enfin, l'élimination de tête admet la compatibilité suivante avec \rightarrow_{η} :

Proposition 4. Soit $c \rightarrow_{\eta} c'$. S'il existe c'' tel que $c' \rightarrow_h c''$, alors $c \rightarrow_h^* c''$.

$$\begin{array}{ccc}
c & \rightarrow_{\eta} & c' \\
& \searrow_{*h} & \downarrow_h \\
& & c''
\end{array}$$

4.2 L'élimination étendue

Définition 11. Si $c \rightarrow_h c'$ alors on définit la relation \rightarrow par, pour tout $C[]$, $C[c] \rightarrow C[c']$, pour une

notion de « commande avec un trou » $C[]$ définie un autre jour.

Proposition 5. Pour toute commande c , on a $c \not\rightarrow$ si et seulement si toute sous-commande c' de c vérifie : c' est une coupure implique c' est malformée.

En particulier, si c est sans coupure, alors $c \not\rightarrow$.

Proposition 6. La relation \rightarrow^* est confluente, i.e. si $c \rightarrow^* c'$ et $c \rightarrow^* c''$, alors il existe c''' tel que $c' \rightarrow^* c'''$ et $c'' \rightarrow^* c'''$.

Enfin, on dispose de la même compatibilité avec \rightarrow_{η} que pour \rightarrow_h :

Proposition 7. Soit $c \rightarrow_{\eta} c'$. S'il existe c'' tel que $C[c'] \rightarrow c''$, alors $C[c] \rightarrow^* c''$.

4.3 Exemples

On pose $\omega = \mu^{\cdot}(x). \langle x \parallel \cdot (x) \rangle$ et $\Omega = \langle \omega \parallel \cdot (\omega) \rangle$. On a alors :

$$\Omega \stackrel{h}{\rightleftharpoons} \langle \omega \parallel \tilde{\mu}x. \langle x \parallel \cdot (x) \rangle \rangle$$

Ce terme n'est pas dérivable dans LL ou LLP, à moins que l'on ne dispose d'une formule P vérifiant $P = !P^{\perp}$ (dans le cas de LLP) ou A vérifiant $A = ?A^{\perp}$ (dans le cas de LL).

Cela invite à poser, lorsque C est une commande avec trou, dans laquelle la variable x n'est ni libre, ni liée :

$$\theta_C = \mu^{\cdot}(x).C \left[\langle x \parallel \cdot (x) \rangle \right]$$

$$\Theta_C = \langle \theta_C \parallel \cdot (\theta_C) \rangle$$

On a alors :

$$\Theta_C \rightarrow_{\mathfrak{h}} \rightarrow_{\mathfrak{h}} C[\Theta_C]$$

Cela se dérive également si l'on dispose d'une formule de la forme ci-dessus, et si C est dérivable.

4.4 Une élimination paresseuse ?

Précisons que lorsqu'il est question d'appel par nom, on entend la stratégie qui duplique les arguments. Ne pas confondre avec l'appel par nécessité, ou encore *appel par valeur paresseux*⁸. La stratégie définie ci-dessus ne permet pas d'implémenter celle-ci. Une évaluation paresseuse au sens de l'appel par nécessité serait basée sur le rejet de l'avarice du μ et de l'automaticité de la réversion.

Remarquer qu'il ne s'agit pas simplement de privilégier systématiquement la réduction du $\tilde{\mu}$. On a en effet vu en introduction que cela mène à un calcul dans lequel on force artificiellement les connecteurs paresseux à être stricts et vice-versa — c'est inutile car on dispose déjà des tous les connecteurs stricts ou paresseux.

5 Réalisabilité

On définit un outil observationnel pour l'étude du système L pur, la *réalisabilité*, basée sur la réalisabilité classique de Krivine pour le λ -calcul [Kri90, Kri93, Kri04].

5.1 Formules

Définition 12 (Formules, formules positives et négatives). On suppose donné un ensemble dénombrable d'*atomes* $\{X, Y \dots\}$ ainsi qu'un ensemble dénombrable de *paramètres* $\{R, S \dots\} \subset \mathcal{P}(\mathcal{V}_+^0)$. On dénote par A, B des formules, par P, Q des formules polarisées positivement et par N, M des formules polarisées négativement. Elles sont engendrées par la grammaire :

$$\begin{aligned} A &::= P \mid N \\ P &::= X \mid R \mid A \otimes A \mid A \oplus A \mid \mathbf{1} \mid \mathbf{0} \mid !A \mid \exists X A \\ N &::= X^\perp \mid R^\perp \mid A \wp A \mid A \& A \mid \perp \mid \top \mid ?A \mid \forall X A \end{aligned}$$

⁸[Her05]. Cf. également la « réduction de tête linéaire » chez Krivine.

Dans la syntaxe ci-dessus, les connecteurs $\forall X$ et $\exists X$ sont liants. On peut ainsi définir les notions habituelles :

- de variables libres $\mathcal{FV}(A)$ d'une formule A , avec pour co-domaine l'ensemble des atomes ;
- de formules closes ;
- d' α -équivalence de formules — toutes les formules seront implicitement prises modulo α -équivalence.

Définition 13 (Dualité). À chaque formule positive P (respectivement chaque formule négative N) correspond une *formule duale* P^\perp négative (resp. N^\perp positive) définie par :

$$\begin{aligned} (X)^\perp &\equiv X^\perp \\ (R)^\perp &\equiv R^\perp \\ (A \otimes B)^\perp &\equiv A^\perp \wp B^\perp \\ (A \oplus B)^\perp &\equiv A^\perp \& B^\perp \\ \mathbf{1}^\perp &\equiv \perp \\ \mathbf{0}^\perp &\equiv \top \\ (\exists X A)^\perp &\equiv \forall X A^\perp \\ (!A)^\perp &\equiv ?(A^\perp) \end{aligned}$$

(resp. :)

$$\begin{aligned} (X^\perp)^\perp &\equiv X \\ (R^\perp)^\perp &\equiv R \\ (A \wp B)^\perp &\equiv A^\perp \otimes B^\perp \\ (A \& B)^\perp &\equiv A^\perp \oplus B^\perp \\ \perp^\perp &\equiv \mathbf{1} \\ \top^\perp &\equiv \mathbf{0} \\ (\forall X A)^\perp &\equiv \exists X A^\perp \\ (?A)^\perp &\equiv !(A^\perp) \end{aligned}$$

On a donc par définition $A^{\perp\perp} = A$ pour toute formule A .

Définition 14. On définit pour toutes formules A, P et tout atome X , la formule A dans laquelle on a substitué P pour X , notée $A[P/X]$, par récurrence sur la structure de A . Les cas importants sont $X[P/X] = P$ et $X^\perp[P/X] = P^\perp$.

5.2 Orthogonal

Définition 15. Pour \mathcal{R} une partie de \mathcal{C}^2 , une partie \perp de \mathcal{C} est dite \mathcal{R} -saturée si :

$$c \mathcal{R} c', c' \in \perp \implies c \in \perp$$

Dans la suite, \perp est une partie \rightarrow_h -saturée et \rightarrow_η -saturée de \mathcal{C}^0 .⁹

Définition 16. Soit C une partie de \mathcal{C}^0 . On pose :

$$\text{Sat}(C) = \left\{ c \in \mathcal{C}^0 \mid \exists c' \in C, c \rightarrow_h^* c' \right\}$$

la clôture \rightarrow_h -saturée de C .

Proposition 8. Si C est \rightarrow_η -saturé, alors $\text{Sat}(C)$ est \rightarrow_h -saturé et \rightarrow_η -saturé.

Démonstration. Cela découle de la compatibilité de \rightarrow_h avec \rightarrow_η (proposition 4). \square

Définition 17.

1. Soit $U \in \mathcal{P}(\mathcal{V}_+^0)$. On pose :

$$U^\perp = \left\{ v_\ominus \in \mathcal{V}_\ominus^0 \mid \forall v_+ \in U, \langle v_+ \parallel v_\ominus \rangle \in \perp \right\}$$

2. Soit $U \in \mathcal{P}(\mathcal{V}_\ominus^0)$. On pose :

$$U^\perp = \left\{ v_+ \in \mathcal{V}_+^0 \mid \forall v_\ominus \in U, \langle v_+ \parallel v_\ominus \rangle \in \perp \right\}$$

3. Une partie de \mathcal{V}_+^0 ou de \mathcal{V}_\ominus^0 est un *comportement* si elle est de la forme U^\perp .

Il y a un conflit de notation ci-dessus, selon que l'on considère \emptyset comme étant une partie de \mathcal{V}_\ominus^0 ou bien comme étant une partie de \mathcal{V}_+^0 . Le contexte permettra alors de décider si l'on a $\emptyset^\perp = \mathcal{V}_+^0$ (premier cas) ou $\emptyset^\perp = \mathcal{V}_\ominus^0$ (second cas).

Proposition 9. Soient U et V deux parties de \mathcal{V}_+^0 ou \mathcal{V}_\ominus^0 .

1. On a $U \subset U^{\perp\perp}$;

2. Si $U \subset V$ alors $V^\perp \subset U^\perp$;

3. On a $U^{\perp\perp\perp} = U^\perp$;

⁹Si l'on souhaite facilement étendre les résultats qui suivent aux calculs des séquents bilatéraux, *i.e.* qui distinguent entre appel par nom et par valeur, il suffit de demander en plus $\langle v \parallel v' \rangle \in \perp \implies \langle v' \parallel v \rangle \in \perp$; ainsi, ce qui suit garde un sens.

4. U est un comportement si et seulement si $U = U^{\perp\perp}$;

5. Si \mathcal{U} est un ensemble de parties de \mathcal{V}_+^0 ou de \mathcal{V}_\ominus^0 , alors on a :

$$\left(\bigcup \mathcal{U} \right)^\perp = \bigcap \left\{ u^\perp \mid u \in \mathcal{U} \right\}$$

6. L'intersection d'une famille quelconque de comportements de même polarité est un comportement.

5.3 Réalisabilité

Définition 18. Soient T et U deux parties de \mathcal{V}_+^0 ou de \mathcal{V}_\ominus^0 . On pose :

$$T \otimes U = \{ (t, u) \mid t \in T, u \in U \}$$

$$T \oplus U = \{ \nu_1(t) \mid t \in T \} \cup \{ \nu_2(u) \mid u \in U \}$$

$$\exists U = \{ \{u\} \mid u \in U \}$$

$$!U = \{ \mu^\lambda(\kappa).c \mid \tilde{\mu}\kappa.c \in U \}$$

(pour κ une variable de polarité opposée à celle de U dans le dernier cas.)

Définition 19. À toute formule close positive P on associe un comportement $|P| \in \mathcal{P}(\mathcal{V}_+^0)$, et à toute formule close négative N on associe un comportement $|N| \in \mathcal{P}(\mathcal{V}_\ominus^0)$. Pour tout terme v , on dit alors que v réalise A , et on écrit $v \Vdash A$, si $v \in |A|$. On procède par récurrence sur la formule considérée :

$$|R^\perp| = R^\perp$$

$$|R| = R^{\perp\perp}$$

$$|A \wp B| = (|A^\perp| \otimes |B^\perp|)^\perp$$

$$|A \otimes B| = (|A| \otimes |B|)^{\perp\perp}$$

$$|A \& B| = (|A^\perp| \oplus |B^\perp|)^\perp$$

$$|A \oplus B| = (|A| \oplus |B|)^{\perp\perp}$$

$$|\perp| = \{()\}^\perp$$

$$|\mathbf{1}| = \{()\}^{\perp\perp}$$

$$|\mathbf{0}| = \{\text{tp}\}^\perp$$

$$|\top| = \{\text{tp}\}^{\perp\perp}$$

$$|?A| = (!|A^\perp|)^\perp$$

$$|!A| = (!|A|)^{\perp\perp}$$

$$|\forall X A| = \left(\bigcup \left\{ \exists |A^\perp[R/X]| \mid R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V}) \right\} \right)^\perp$$

$$|\exists X A| = \left(\bigcup \left\{ \exists |A[R/X]| \mid R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V}) \right\} \right)^{\perp\perp}$$

Proposition 10. Pour toute formule A , on a :

$$|A|^\perp = |A^\perp|$$

Corollaire 1. Soient v un terme clos et A une formule close. On a $v \Vdash A$ si et seulement si $\forall v' (v' \Vdash A^\perp \Rightarrow \langle v \parallel v' \rangle \in \perp)$.

Corollaire 2. Soit A une formule ayant X pour seule variable libre et P une formule positive close. On a :

$$|A[|P|/X]| = |A[P/X]|$$

Donc aussi, pour toute formule négative close N :

$$|A[|N|^\perp/X]| = |A[N^\perp/X]|$$

Remarque 4. Toute formule P est réalisée si \perp n'est pas vide. En effet, si $c \in \perp$, on a $\mu_{_} c \Vdash P$, où $_$ est une variable négative libre dans c . En revanche, si $\perp = \emptyset$, il existe des formules non réalisées, comme \perp ou $\mathbf{0}$. Cela montre en particulier que les systèmes qui ont leur lemme d'adéquation sont cohérents.

5.4 Théorème de génération

Définition 20. Soient U un comportement et V une partie de U . On dit que V est un générateur de (ou engendre) U si $V^\perp \subset U^\perp$.

Proposition 11. Soit A une formule telle que $\mathcal{FV}(A) = \{X\}$. Soit U un comportement et V un générateur de U . On a $|A[U/X]| = |A[V/X]|$.

Démonstration. Par récurrence immédiate, essentiellement car $|V| = V^{\perp\perp} = U$.

Définition 21. Soit U un comportement. L'ensemble $U_{\mathbb{V}}$ des valeurs de U est l'intersection de U et de l'ensemble des valeurs \mathbb{V} .

□

Théorème 1 (Génération). Si A est une formule close sans paramètre, alors $|A|$ est engendré par l'ensemble de ses valeurs.

Démonstration. Il s'agit d'une conséquence du lemme 4 qui suit. □

Lemme 1. Soit U une partie de \mathcal{V}_+^0 (resp. \mathcal{V}_\ominus^0) et $c \in \mathcal{C}$ avec $\mathcal{FV}(c) = \{\kappa\}$ avec κ une variable positive (resp. négative). Si pour tout $V \in U_{\mathbb{V}}$, on a $c[V/\kappa] \in \perp$, alors $\tilde{\mu}\kappa.c \in U_{\mathbb{V}}^\perp$.

Démonstration. Soit $V \in U_{\mathbb{V}}$. On a $\langle V \parallel \tilde{\mu}\kappa.c \rangle (\rightarrow_{\tilde{\mu}} U \rightarrow_{\mu}) c[V/\kappa] \in \perp$, donc $\langle V \parallel \tilde{\mu}\kappa.c \rangle \in \perp$ car \perp est \rightarrow_h -saturé. D'où le résultat. □

Lemme 2. Les propriétés suivantes sont vérifiées :

1. Si une partie U de \mathcal{V}_+^0 ou de \mathcal{V}_\ominus^0 est telle que $U_{\mathbb{V}}^\perp \subset U^\perp$, alors $U^{\perp\perp}$ est engendré par l'ensemble de ses valeurs.
2. Si un comportement U est engendré par un ensemble de valeurs, alors U est engendré par l'ensemble de ses valeurs.
3. Si U est un comportement, alors :

$$U_{\mathbb{V}}^{\perp\perp} \cap \mathbb{V} = U_{\mathbb{V}}$$

Démonstration.

1. On a $U_{\mathbb{V}} \subset U^{\perp\perp} \cap \mathbb{V}$, donc $U^{\perp\perp}_{\mathbb{V}} \subset U_{\mathbb{V}}^\perp \subset U^\perp$.
2. Conséquence de 1.
3. (\supset) Trivial. (\subset) U étant un comportement, le membre de droite s'écrit $U^{\perp\perp} \cap \mathbb{V}$. Or $U_{\mathbb{V}}^{\perp\perp} \subset U^{\perp\perp}$, d'où le résultat.

□

Lemme 3. Soit U et V deux comportements. Les propriétés suivantes sont vérifiées :

1. $U_{\mathbb{V}}^{\perp\perp} \otimes V_{\mathbb{V}}^{\perp\perp} \subset (U \otimes V)_{\mathbb{V}}^{\perp\perp}$;
2. $U_{\mathbb{V}}^{\perp\perp} \oplus V_{\mathbb{V}}^{\perp\perp} \subset (U \oplus V)_{\mathbb{V}}^{\perp\perp}$;
3. $\exists(U_{\mathbb{V}}^{\perp\perp}) \subset (\exists U)_{\mathbb{V}}^{\perp\perp}$.

Démonstration.

1. Par définition de l'ensemble des valeurs, on a $U_{\mathbb{V}} \otimes V_{\mathbb{V}} = (U \otimes V)_{\mathbb{V}}$. Il suffit donc de montrer $(U_{\mathbb{V}} \otimes V_{\mathbb{V}})^\perp \subset (U_{\mathbb{V}}^{\perp\perp} \otimes V_{\mathbb{V}}^{\perp\perp})^\perp$. Soit $v_\ominus \in (U_{\mathbb{V}} \otimes V_{\mathbb{V}})^\perp$; soit $u \in U_{\mathbb{V}}^{\perp\perp}$ et $v \in V_{\mathbb{V}}^{\perp\perp}$. Par \rightarrow_η -saturation de \perp , pour des variables κ et κ' de polarités respectivement celles de u et de v , il suffit de montrer $\langle (u, v) \parallel \tilde{\mu}(\kappa, \kappa'). \langle (\kappa, \kappa') \parallel v_\ominus \rangle \rangle \in \perp$. Mais pour tous $u' \in U_{\mathbb{V}}$ et $v' \in V_{\mathbb{V}}$, par définition de v_\ominus , on a $\langle (u', v') \parallel v_\ominus \rangle \in \perp$. Donc $\tilde{\mu}\kappa'. \langle (u', \kappa') \parallel v_\ominus \rangle \in V_{\mathbb{V}}^\perp$ par le lemme 1. Donc aussi $\langle v \parallel \tilde{\mu}\kappa'. \langle (u', \kappa') \parallel v_\ominus \rangle \rangle \in$

\perp par hypothèse sur v . Donc $\tilde{\mu}\kappa.\langle v \parallel \tilde{\mu}\kappa'.\langle(\kappa, \kappa') \parallel v_\ominus\rangle\rangle \in \mathcal{U}_V^\perp$ par le lemme 1, d'où le résultat, puisque :

$$\begin{aligned} & \langle (u, v) \parallel \tilde{\mu}(\kappa, \kappa').\langle(\kappa, \kappa') \parallel v_\ominus\rangle \rangle \rightarrow_h \\ & \langle u \parallel \tilde{\mu}\kappa.\langle v \parallel \tilde{\mu}\kappa'.\langle(\kappa, \kappa') \parallel v_\ominus\rangle\rangle \rangle \in \perp \end{aligned}$$

2. Même raisonnement que 1.

3. Même raisonnement que 2. \square

Lemme 4. Soit A une formule close dont tout paramètre R est tel que $R_V^\perp \subset R^\perp$. Alors $|A|_V$ est un générateur de $|A|$.

Démonstration. Par récurrence sur la taille de la formule A .

- Si A est négatif, alors le résultat est trivial car tout terme négatif est une valeur.
- Si A est un paramètre R , alors $|A| = R^{\perp\perp}$. Par hypothèse, on a $R_V^\perp \subset R^\perp = |A|^\perp$, donc par le lemme 2, $|A|$ est engendré par l'ensemble de ses valeurs ;
- Cas où $A = B \otimes C$. Par hypothèse de récurrence, $|B| \otimes |C|$ est égal à $|B|_V^{\perp\perp} \otimes |C|_V^{\perp\perp}$, lequel est inclus dans $(|B| \otimes |C|)_V^{\perp\perp}$ par le lemme 3. Donc *a fortiori* $|A|_V^\perp \subset (|B| \otimes |C|)_V^\perp \subset (|B| \otimes |C|)^\perp = |A|^\perp$.
- Cas où $A = B \oplus C$. Même preuve en remplaçant \otimes par \oplus .
- Cas où $A = !B$. $|A|$ est de la forme $(!|B|)^{\perp\perp}$ avec $!|B|$ un ensemble de valeurs ; $|A|$ est donc engendré par ses valeurs selon le lemme 2.
- Cas où $A = \mathbf{1}$. On a $\mathbf{1} = \{()\}^{\perp\perp}$ avec $()$ une valeur ; donc $\mathbf{1}$ est engendré par ses valeurs selon le lemme 2.
- Cas où $A = \exists X B$. On a par hypothèse de récurrence, pour tout $R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V})$, $|B[R/X]|_V^\perp \subset |B[R/X]|^\perp$, et donc, d'après le lemme 3, $(\exists B[R/X])_V^\perp \subset (\exists B[R/X])^\perp$. Donc aussi :

$$\bigcap_{R \subset \mathcal{V}_+^0 \cap \mathbb{V}} (\exists B[R/X])_V^\perp = \bigcap_{R \subset \mathcal{V}_+^0 \cap \mathbb{V}} (\exists B[R/X])^\perp$$

D'où, par la proposition 9.5, $|\exists X B|_V^\perp \subset |\exists X B|^\perp$. \square

Remarque 5. La démonstration pour le cas de la quantification au second ordre reste valide si l'on se passe du constructeur d'extraction $\{\cdot\}$ pour dénoter l'introduction du \exists . En revanche, en l'absence d'un constructeur dans la syntaxe pour échanger la polarité des termes, $\forall X P$ doit être un positif, ce qui pose des problèmes pour le lemme ci-dessus. En effet, la définition de $|\forall X P|$ est alors :

$$\bigcap \left\{ |P[R/X]| \mid R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V}) \right\}$$

Or, le contre-exemple qui suit montre qu'une intersection de comportements engendrés par leurs valeurs n'est pas forcément engendrée par ses valeurs.

En revanche, dans les systèmes polarisés, qui interdisent donc $\forall X P$, comme LLP, cela signifie qu'il n'y a pas d'objection à employer une syntaxe « à la Krivine », i.e. sans constructeur pour dénoter le second-ordre.

Proposition 12. L'intersection de deux comportements engendrés par leurs valeurs n'est pas nécessairement engendrée par ses valeurs. *A fortiori*, un comportement quelconque n'est pas nécessairement engendré par l'ensemble de ses valeurs.

Démonstration. En voici un contre-exemple. On pose :

$$\perp = \text{Sat} \left(\left\{ \langle \iota_1() \parallel e_1 \rangle, \langle \iota_2() \parallel e_2 \rangle, \langle v_1 \parallel e_1 \rangle, \langle v_1 \parallel e_2 \rangle \right\} \right)$$

où $v_1 = \mu_-. \Omega$, $e_1 = \tilde{\mu}_-. \Omega$ et $e_2 = \tilde{\mu}_-. \langle () \parallel \tilde{\mu}_-. \Omega \rangle$. \perp est \rightarrow_h - et \rightarrow_η -saturé car il est la clôture \rightarrow_h -saturée d'un ensemble \rightarrow_η -saturé. On pose $\mathcal{U} = |\mathbf{1} \oplus \mathbf{0}| \cap |\mathbf{0} \oplus \mathbf{1}| = \{\iota_1()\}^{\perp\perp} \cap \{\iota_2()\}^{\perp\perp}$. \mathcal{U} n'est pas engendré par l'ensemble de ses valeurs : en effet, on remarque tout d'abord que $v_1 \in \mathcal{U}$ n'appartient pas à $\emptyset^{\perp\perp}$, donc $\mathcal{U} \neq \emptyset^{\perp\perp}$. Pourtant, on a $\mathcal{U}_V = \emptyset$. En effet, si l'on suppose qu'il existe une valeur $V \in \mathcal{U}$, une telle valeur vérifie à la fois $\langle V \parallel e_1 \rangle \in \perp$ et $\langle V \parallel e_2 \rangle \in \perp$, ce qui est absurde. \square

Corollaire 3. Soit A une formule close et $c \in \mathcal{C}$ avec $\mathcal{FV}(c) = \{\kappa\}$ avec κ une variable de la polarité de A . Pour montrer $\tilde{\mu}\kappa.c \Vdash A^\perp$, il suffit de montrer : pour tout $V \in |A|_V$, on a $c[V/\kappa] \in \perp$.

6 La logique linéaire du second ordre

Définition 22. Un *contexte* est un ensemble d'éléments de la forme $x : N$ ou $\alpha : P$.

Définition 23. Les *séquents* du système L sont des jugements de la forme suivante :

$$\begin{array}{l} c : (\vdash \Gamma) \\ \vdash v_+ : P; \Gamma \\ \vdash v_\ominus : N; \Gamma \end{array}$$

où Γ est un contexte.

Dans $\vdash v_+ : P; \Gamma$ (resp. dans $\vdash v_\ominus : N; \Gamma$), la formule P (resp. N) est dite *activée*. Cela correspond à peu de choses près à la notion de *formule principale*. Ne pas confondre avec celle de *bénéficiaire* ou celle de *formule focalisée*.

La logique linéaire du second ordre est définie fig. 4.

Dans le groupe logique, les règles d'activation, ainsi que les règles de désactivation qui suivent, énoncent l'équivalence entre les commandes $c : (\vdash A, \Gamma)$ et les termes $\vdash v : A, \Gamma$.

Proposition 13. On dérive du groupe identité les règles suivantes :

$$\frac{\vdash v_\ominus : N; \Gamma}{\langle x \parallel v_\ominus \rangle : (\vdash x : N, \Gamma)} \text{ (désactivation } \ominus)$$

$$\frac{\vdash v_+ : P; \Gamma}{\langle v_+ \parallel \alpha \rangle : (\vdash \Gamma, \alpha : P)} \text{ (désactivation } +)$$

Remarque 6. On peut maintenant justifier notre notation pour la constante tp . On pose $\mathcal{A}(v_+) \equiv \mu_{\cdot} \langle v_+ \parallel tp \rangle$ — si tp est la « constante *toplevel* », alors il s'agit d'une définition de l'opérateur *abort* similaire à celle d'Ariola, Herbelin et Sabry [AHS04] pour le calcul $\lambda\mu$. De la règle \top est maintenant conséquence la règle suivante :

$$\frac{\vdash v_+ : \mathbf{0}; \Gamma}{\vdash \mathcal{A}(v_+) : P; \Gamma} \text{ (Ex falso quodlibet)}$$

6.1 Lemme d'adéquation

Théorème 2 (Lemme d'adéquation – LL). *Soit c une commande (respectivement v un terme) typable dans LL, de type $\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n$ (resp. $\vdash v : A; \kappa_1 : A_1, \dots, \kappa_n : A_n$) avec A_1, \dots, A_n (resp. et A) des formules closes. Alors pour tous termes clos v_1, \dots, v_n , si $v_1 \Vdash A_1^\perp, \dots, v_n \Vdash A_n^\perp$, alors $c[v_1/\kappa_1, \dots, v_n/\kappa_n] \in \perp$ (resp. $v[v_1/\kappa_1, \dots, v_n/\kappa_n] \Vdash A$).*

Démonstration. Il s'agit d'une conséquence du lemme qui suit. \square

Lemme 5. *Soit $c \in \mathcal{C}$ (respectivement $v \in \mathcal{V}_+^0 \cup \mathcal{V}_\ominus^0$) typable dans LL, de type $\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n$ (resp. de type $\vdash v : B; \kappa_1 : A_1, \dots, \kappa_n : A_n$). Ces formules ont X_1, \dots, X_m pour variables libres. Soit des paramètres $R_1, \dots, R_m \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathcal{V})$ et des termes clos v_1, \dots, v_n . On note $[\vec{v}_i/\vec{\kappa}_i]$ la substitution $[v_1/\kappa_1, \dots, v_n/\kappa_n]$ et $[\vec{R}_j/\vec{X}_j]$ la substitution $[R_1/X_1, \dots, R_m/X_m]$. On pose pour tout $i \leq n$, $A'_i = A_i[\vec{R}_j/\vec{X}_j]$.*

Si $v_1 \Vdash A'_1, \dots, v_n \Vdash A'_n$, alors $c[\vec{v}_i/\vec{\kappa}_i] \in \perp$ (resp. $v[\vec{v}_i/\vec{\kappa}_i] \Vdash B'$ où $B' = B[\vec{R}_j/\vec{X}_j]$).

Démonstration. Par récurrence sur la dérivation de c et de v :

Axiome Les hypothèses sont $\vdash \kappa_1 : A_1; \kappa_1 : A_1$ et $v_1 \Vdash A_1^\perp$. On a bien $\kappa_1[v_1/\kappa_1] \Vdash A_1^\perp$.

Activation $\vdash \tilde{\mu}\kappa.c : B; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $c : (\vdash \kappa : B; \kappa_1 : A_1, \dots, \kappa_n : A_n)$. Pour montrer $(\tilde{\mu}\kappa.c)[\vec{v}_i/\vec{\kappa}_i] \Vdash B'$, il suffit de montrer, par le théorème 1 que pour tout $v \in |\mathcal{B}^\perp|_{\mathcal{V}}$, on a $c[\vec{v}_i/\vec{\kappa}_i][v/\kappa] \in \perp$: cela est fourni par l'hypothèse de récurrence.

Coupure $\langle v \parallel v' \rangle : (\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n)$ provient de $\vdash v : B; \kappa_1 : A_1, \dots, \kappa_i : A_i$ et de $\vdash v' : B^\perp; \kappa_{i+1} : A_{i+1}, \dots, \kappa_n : A_n$, avec $0 \leq i \leq n$. On a par hypothèse de récurrence $v[v_1/\kappa_1, \dots, v_i/\kappa_i] \Vdash B'$ et $v'[v_{i+1}/\kappa_{i+1}, \dots, v_n/\kappa_n] \Vdash B'^\perp$. Puisque $|\mathcal{B}^\perp[\vec{R}_j/\vec{X}_j]| = |\mathcal{B}[\vec{R}_j/\vec{X}_j]|^\perp$, on a bien $\langle v \parallel v' \rangle[\vec{v}_i/\vec{\kappa}_i] \in \perp$.

Déréliction $\vdash \backslash(v) : ?B; \kappa_1 : A_1, \dots, \kappa_n : A_n$ vient de $\vdash v : B; \kappa_1 : A_1, \dots, \kappa_n : A_n$. On pose $w = v[\vec{v}_i/\vec{\kappa}_i]$. Pour obtenir $\backslash(w) \Vdash ?B'$, il suffit, par définition de $|\mathcal{B}'|$, de montrer, pour $v' \Vdash B'^\perp$ quelconque et κ de la polarité de B' , $\langle \mu \backslash(\kappa) \cdot \langle v' \parallel \kappa \rangle \parallel \backslash(w) \rangle \in \perp$. Par

Groupe identité

Règles axiome, activation, coupure :

$$\frac{}{\vdash x : P; x : P^\perp} \text{ (ax+)} \quad \frac{}{\vdash \alpha : N; \alpha : N^\perp} \text{ (ax}\ominus\text{)}$$

$$\frac{c : (\vdash x : N, \Gamma)}{\vdash \tilde{\mu}x.c : N; \Gamma} \text{ (}\tilde{\mu}\text{)} \quad \frac{c : (\vdash \Gamma, \alpha : P)}{\vdash \mu\alpha.c : P; \Gamma} \text{ (}\mu\text{)}$$

$$\frac{\vdash v : A; \Gamma \quad \vdash v' : A^\perp; \Delta}{\langle v \parallel v' \rangle : (\vdash \Gamma, \Delta)} \text{ (cut)}$$

Groupe structurel

Règles promotion, déréluction, affaiblissement, contraction :

$$\frac{\vdash v : A; \Gamma}{\vdash \lambda(v) : ?A; \Gamma} \text{ (?d)} \quad \frac{c : (\vdash \kappa : A, ?\Gamma)}{\vdash \mu^\lambda(\kappa).c : !A; ?\Gamma} \text{ (!)}$$

$$\frac{c : (\vdash \Gamma)}{c : (\vdash x : ?A, \Gamma)} \text{ (?w)} \quad \frac{c : (\vdash x : ?A, y : ?A, \Gamma)}{c[x/y] : (\vdash x : ?A, \Gamma)} \text{ (?c)}$$

$$\frac{\vdash v : A; \Gamma}{\vdash v : A; x : ?B, \Gamma} \text{ (?w)} \quad \frac{\vdash v : A; x : ?B, y : ?B, \Gamma}{\vdash v[x/y] : A; x : ?B, \Gamma} \text{ (?c)}$$

Groupe logique

Règles tenseur, par, avec, plus :

$$\frac{\vdash v : A; \Gamma \quad \vdash v' : B; \Delta}{\vdash (v, v') : A \otimes B; \Gamma, \Delta} \text{ (}\otimes\text{)} \quad \frac{c : (\vdash \kappa : A, \kappa' : B, \Gamma)}{\vdash \tilde{\mu}(\kappa, \kappa').c : A \wp B; \Gamma} \text{ (}\wp\text{)}$$

$$\frac{c : (\vdash \kappa : A, \Gamma) \quad c' : (\vdash \kappa' : B, \Gamma)}{\vdash \tilde{\mu}(\iota_1(\kappa).c \mid \iota_2(\kappa').c') : A \& B; \Gamma} \text{ (&)} \quad \frac{\vdash v : A_i; \Gamma}{\vdash \iota_i(v) : A_1 \oplus A_2; \Gamma} \text{ (}\oplus_{i \in \{1,2\}}\text{)}$$

Extraction, généralisation :

$$\frac{\vdash v : A[P/X]; \Gamma}{\vdash \{v\} : \exists X A; \Gamma} \text{ (}\exists\text{)} \quad \frac{c : (\vdash \kappa : A, \Gamma)}{\vdash \tilde{\mu}\{\kappa\}.c : \forall X A; \Gamma} \text{ (}\forall\text{)} \text{ (}X \notin \mathcal{FV}(\Gamma)\text{)}$$

Règles unités :

$$\frac{}{\vdash () : \mathbf{1};} \text{ (}\mathbf{1}\text{)} \quad \frac{c : (\vdash \Gamma)}{\vdash \tilde{\mu}().c : \perp; \Gamma} \perp$$

$$\frac{}{\vdash \text{tp} : \top; \Gamma} \text{ (}\top\text{)} \quad \text{pas de règle pour } \mathbf{0}$$

FIG. 4: La logique linéaire du second ordre.

\rightarrow_h -saturation, cela se réduit à $\langle v \parallel w \rangle \in \perp$, ce qui est vérifié car par hypothèse de récurrence on a $w \Vdash B'$.

Promotion $\vdash \mu^\lambda(\kappa).c : !B; \kappa_1 : ?A_1, \dots, \kappa_n : ?A_n$ provient de $c : (\vdash \kappa : B, \kappa_1 : ?A_1, \dots, \kappa_n : ?A_n)$. On a su montrer plus haut que $(\tilde{\mu}\kappa.c)[\vec{v}_i/\vec{\kappa}_i] \Vdash B[\vec{R}_j/\vec{X}_j]$.

On a donc aussi $(\mu^\lambda(\kappa).c)[\vec{v}_i/\vec{\kappa}_i] \Vdash !B[\vec{R}_j/\vec{X}_j]$.

Affaiblissement et contraction Trivial.

Tenseur $\vdash (t, u) : B \otimes C; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $\vdash t : B; \kappa_1 : A_1, \dots, \kappa_i : A_i$ et de $\vdash u : C; \kappa_{i+1} : A_{i+1}, \dots, \kappa_n : A_n$, avec $0 \leq i \leq n$. Le résultat suit de ce que $|B'| \otimes |C'| \subset |B' \otimes C'|$.

Plus $\vdash \iota_i(t) : B_1 \oplus B_2; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $\vdash t : B_i; \kappa_1 : A_1, \dots, \kappa_n : A_n$, avec $i \in \{1, 2\}$. Le résultat découle de $\{\iota_i(t) \mid t \in |B'_i|\} \subset |B'_1 \oplus B'_2|$.

Par $\vdash \tilde{\mu}(\kappa, \kappa').c : B \wp C; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $c : (\vdash \kappa : B, \kappa' : C, \kappa_1 : A_1, \dots, \kappa_n : A_n)$. Il suffit, pour montrer $\tilde{\mu}(\kappa, \kappa').c[\vec{v}_i/\vec{\kappa}_i] \in (|B'^\perp| \otimes |C'^\perp|)^\perp$, de montrer $\tilde{\mu}(\kappa, \kappa').c[\vec{v}_i/\vec{\kappa}_i] \in (|B'^\perp|_{\mathbb{V}} \otimes |C'^\perp|_{\mathbb{V}})^\perp$, par le théorème 1. Or, pour $(t, u) \in |B'^\perp|_{\mathbb{V}} \otimes |C'^\perp|_{\mathbb{V}}$, on a :

$$\langle (t, u) \parallel \tilde{\mu}(\kappa, \kappa').c[\vec{v}_i/\vec{\kappa}_i] \rangle \rightarrow_h^* c[t/\kappa, u/\kappa', \vec{v}_i/\vec{\kappa}_i]$$

car t et u sont des valeurs. D'où le résultat par hypothèse de récurrence.

Avec $\vdash \tilde{\mu}(\iota_1(\kappa'_1).c_1 \mid \iota_2(\kappa'_2).c_2) : B_1 \& B_2; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $c_1 : (\vdash \kappa'_1 : B_1; \kappa_1 : A_1, \dots, \kappa_n : A_n)$ et de $c_2 : (\vdash \kappa'_2 : B_2; \kappa_1 : A_1, \dots, \kappa_n : A_n)$. Par le théorème 1., on a $(|B'_1|^\perp \oplus |B'_2|^\perp)^\perp \subset (|B'_1|_{\mathbb{V}}^\perp \oplus |B'_2|_{\mathbb{V}}^\perp)^\perp$. Soit alors $\iota_i(t) \in |B'_1|_{\mathbb{V}}^\perp \oplus |B'_2|_{\mathbb{V}}^\perp$. On a :

$$\langle \iota_i(t) \parallel \tilde{\mu}(\iota_1(\kappa'_1).c_1 \mid \iota_2(\kappa'_2).c_2)[\vec{v}_i/\vec{\kappa}_i] \rangle \rightarrow_h^2 c_i[t/\kappa'_i, \vec{v}_i/\vec{\kappa}_i]$$

car t est une valeur. D'où le résultat par hypothèse de récurrence.

Extraction $\vdash \{t\} : \exists X B; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $\vdash t : B[P/X]; \kappa_1 : A_1, \dots, \kappa_n : A_n$, avec X distinct des X_1, \dots, X_m .

On pose $P' = P[\vec{R}_j/\vec{X}_j]$. Par le théorème 1, on a $|P'|_{\mathbb{V}}^\perp = |P'|^\perp$, donc $|B'[P'/X]| = |B'|[|P'|_{\mathbb{V}}/X]|$. Or, par hypothèse de récurrence, on a $t[\vec{v}_i/\vec{\kappa}_i] \Vdash B[P/X][\vec{R}_j/\vec{X}_j]$. On a donc :

$$\{t\}[\vec{v}_i/\vec{\kappa}_i] \in \exists |B'|[|P'|_{\mathbb{V}}/X]|$$

Puisque $|P'|_{\mathbb{V}}$ est bien une partie de $\mathcal{V}_+^0 \cap \mathbb{V}$, on a aussi :

$$\{t\}[\vec{v}_i/\vec{\kappa}_i] \Vdash \exists X B'$$

Généralisation $\vdash \tilde{\mu}\{\kappa\}.c : \forall X B; \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $c : (\vdash \kappa : B; \kappa_1 : A_1, \dots, \kappa_n : A_n)$, avec X distinct des X_1, \dots, X_m . Par la proposition 9.5, on a :

$$|\forall X B'| = \bigcap_{R \subset \mathcal{V}_+^0 \cap \mathbb{V}} (\exists |B'[R/X]|)^\perp$$

Soit $R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V})$. Par le théorème 1, il suffit de montrer :

$$\tilde{\mu}\{\kappa\}.c[\vec{v}_i/\vec{\kappa}_i] \in (\exists |B'[R/X]|_{\mathbb{V}})^\perp$$

Soit alors $\{t\} \in \exists |B'[R/X]|_{\mathbb{V}}$. On a :

$$\langle \{t\} \parallel \tilde{\mu}\{\kappa\}.c \rangle \rightarrow_h^2 c[t/\kappa, \vec{v}_i/\vec{\kappa}_i]$$

car t est une valeur. D'où le résultat par hypothèse de récurrence, puisque $t \in |B[R/X, \vec{R}_j/\vec{X}_j]|$.

1, \perp et \top Trivial. □

Remarque 7. Remarquer que la réalisabilité n'est pas impliquée vis-à-vis de la linéarité : on pourrait par exemple ajouter des contractions et des affaiblissements gratuits sans qu'elle n'y trouve à redire. C'est un outil du niveau -2 , qui parle de ce qu'on peut observer de la syntaxe, sans se soucier de la procéduralité.

Le lemme d'adéquation possède les corollaires suivants :

Corollaire 4 (Normalisation de tête). *Soit P une formule positive et $t \in \mathcal{V}_+^0$ tel que $\vdash t : P$. Alors il existe une valeur $V \in \mathcal{V}_+^0$ tel que :*

$$\langle t \parallel tp \rangle \rightarrow_h^* \langle V \parallel tp \rangle$$

Démonstration. On pose $\perp = \text{Sat} \left\{ \langle V \parallel tp \mid V \in \mathcal{V}_+^0 \cap \mathbb{V} \rangle \right\}$, qui est \rightarrow_{η} -saturé car $\left\{ \langle V \parallel tp \mid V \in \mathcal{V}_+^0 \cap \mathbb{V} \rangle \right\}$ l'est. On a pour tout $V \in |\mathbb{P}|_{\mathbb{V}}$, $\langle V \parallel tp \rangle \in \perp$, c'est-à-dire $tp \in |\mathbb{P}|_{\mathbb{V}}^{\perp}$, et donc $tp \Vdash P^{\perp}$ par le théorème 1. Or, par le lemme d'adéquation, on a $t \Vdash P$. D'où $\langle t \parallel tp \rangle \in \perp$, ce qu'il faut démontrer. \square

Corollaire 5 (Propriété de la disjonction). *Soit une formule de la forme $A_1 \oplus A_2$ et $t \in \mathcal{V}_+^0$ tel que $\vdash t : A_1 \oplus A_2$. Alors il existe $i \in \{1, 2\}$ et une valeur close V de la polarité de A_i tel que :*

$$\langle t \parallel tp \rangle \rightarrow_h^* \langle \iota_i(V) \parallel tp \rangle$$

Démonstration. On pose E l'ensemble des $\langle \iota_i(V) \parallel tp \rangle$ avec $i \in \{1, 2\}$ et V une valeur close de la polarité de A_i . On pose $\perp = \text{Sat}(E)$, qui est \rightarrow_{η} -saturé car E l'est. On a pour tout $V \in |A_i|_{\mathbb{V}}$, $\langle \iota_i(V) \parallel tp \rangle \in \perp$, c'est-à-dire $tp \in |A_1 \oplus A_2|_{\mathbb{V}}^{\perp}$, et donc $tp \Vdash A_1^{\perp} \& A_2^{\perp}$ par le théorème 1. Or, par le lemme d'adéquation, on a $t \Vdash A_1 \oplus A_2$. D'où $\langle t \parallel tp \rangle \in \perp$, ce qu'il faut démontrer. \square

Exemple 1. Soit t un terme typable dans LL de type¹⁰ $\vdash t : \forall X(X \multimap X)$ et V une valeur positive. On a

$$\langle t \parallel \{(V, tp)\} \rangle \rightarrow_h^* \langle V \parallel tp \rangle$$

En effet, on pose $\perp = \text{Sat} \{ \langle V \parallel tp \rangle \}$, qui est \rightarrow_{η} -saturé. On a ainsi $\langle V, tp \rangle \Vdash \{V\} \otimes \{V\}^{\perp}$, donc aussi $\{(V, tp)\} \Vdash \exists X(X \otimes X^{\perp})$, puisque $\{V\}$ est un ensemble de valeurs. Comme $t \Vdash \forall X(X \multimap X)$ par le lemme d'adéquation, on a $\langle t \parallel \{(V, tp)\} \rangle \in \perp$, d'où le résultat.

Exemple 2. Soit t un terme typable dans LL de type $\vdash t : \forall X(X \otimes X \multimap X \otimes X)$ et V_1 et V_2 deux valeurs positives. On a :

$$\langle t \parallel \{((V_1, V_2), tp)\} \rangle \rightarrow_h^* \langle (V_1, V_2) \parallel tp \rangle$$

ou :

$$\langle t \parallel \{((V_1, V_2), tp)\} \rangle \rightarrow_h^* \langle (V_2, V_1) \parallel tp \rangle$$

En effet, on pose :

$$\perp = \text{Sat} \left\{ \langle (V_i, V_j) \parallel tp \mid i, j \in \{1, 2\} \rangle \right\}$$

¹⁰ $A \multimap B$ est défini par $A^{\perp} \wp B$.

qui est \rightarrow_{η} -saturé. On pose $R = \{V_1, V_2\}$. On a ainsi $tp \Vdash R^{\perp} \wp R^{\perp}$, donc aussi :

$$\langle (V_1, V_2), tp \rangle \Vdash (R \otimes R) \otimes (R^{\perp} \wp R^{\perp})$$

Comme R est un ensemble de valeurs, on a :

$$\langle \{(V_1, V_2), tp\} \rangle \Vdash \exists X((X \otimes X) \otimes (X^{\perp} \wp X^{\perp}))$$

Comme $t \Vdash \forall X(X \otimes X \multimap X \otimes X)$ par le lemme d'adéquation, on a $\langle t \parallel \{((V_1, V_2), tp)\} \rangle \in \perp$.

Pour obtenir le résultat, il est encore nécessaire de montrer que l'on n'obtient ni $\langle (V_1, V_1) \parallel tp \rangle$ ni $\langle (V_2, V_2) \parallel tp \rangle$. Cela découle de la linéarité de t , mais la réalisabilité n'est pas impliquée dans les questions de procéduralité. C'est un autre outil qu'il faut donc employer ici.

6.2 Atomes sans polarité

Le quotient de la logique linéaire présenté ici est le seul qui soit à la fois aussi complet et simple. La seule license que l'on s'est permis vis-à-vis du calcul original concerne la polarité des atomes : ici, un atome X est positif tandis que X^{\perp} est négatif, ce qui est une restriction qui n'existe pas dans le calcul des séquents original. Toutes les formules non-atomique possédant une polarité, cette contrainte sur les atomes est légitime. On montre qu'elle est également superficielle.

Dans la logique linéaire, $\forall X(X \multimap X)$ représente à la fois $\forall X(X \multimap X)$ et $\forall X(X^{\perp} \multimap X^{\perp})$ dans notre système, par exemple. La restriction n'est pas grande : pour obtenir le même pouvoir expressif, il suffit de considérer, de façon informelle, des « atomes neutres », disons $X^?$. Une preuve de $\forall X(X^? \multimap X^?)$ sera obtenue lorsque l'on pourra à la fois prouver $\forall X(X \multimap X)$ et $\forall X(X^{\perp} \multimap X^{\perp})$.

La seule différence, alors, au niveau des termes de preuve obtenus, apparaît si ces preuves utilisent la règle axiome pour $X^?$. Cette règle est de la forme suivante : $\vdash \kappa : X^?; \kappa : X^{\perp}$. Cela signifie qu'avec la convention pour laquelle κ, κ', \dots dénotent des variables qui peuvent au choix être positives ou négatives, les deux preuves de $\forall X(X \multimap X)$ et $\forall X(X^{\perp} \multimap X^{\perp})$ sont informellement représentés par un unique terme écrit avec des κ, κ', \dots .

On observe que de telles variables « neutres » obéissent à certaines contraintes dans le cadre de LL :

- Elles sont linéaires ;
- Si $\tilde{\mu}\kappa.c$ apparaît dans une coupure, alors son contre-terme est de la forme $\tilde{\mu}\kappa'.c'$ ou κ' .

Exemple 3. On considère les termes de preuve suivant :

- Le morphisme $m : (\vdash \alpha : A^\perp \oplus A^\perp, \beta : A \oplus A)$ avec :

$$m = \left\langle \tilde{\mu}\kappa.\langle \alpha \parallel (\kappa \mid \kappa) \rangle \parallel \tilde{\mu}\kappa'.\langle (\kappa' \mid \kappa') \parallel \beta \rangle \right\rangle$$

- La « dissociativité¹¹ » :

$d : (\vdash \kappa' : A^\perp, \alpha : B^\perp \otimes C^\perp, \gamma : A \otimes B, \kappa'' : C)$
avec :

$$d = \left\langle \tilde{\mu}\kappa.\langle \alpha \parallel (\kappa, \kappa'') \rangle \parallel \tilde{\mu}\kappa.\langle (\kappa', \kappa) \parallel \gamma \rangle \right\rangle$$

On observe le cas de cette dissociativité.

- Si l'on suppose B positif, alors d s'écrit :

$$\left\langle \mu\beta.\langle \alpha \parallel (\beta, \kappa'') \rangle \parallel \tilde{\mu}x.\langle (\kappa', x) \parallel \gamma \rangle \right\rangle$$

Et on a donc :

$$d \rightarrow_h \left\langle \alpha \parallel (\tilde{\mu}x.\langle (\kappa', x) \parallel \gamma \rangle, \kappa'') \right\rangle$$

- Si l'on suppose B négatif, alors d s'écrit :

$$\left\langle \tilde{\mu}x.\langle \alpha \parallel (x, \kappa'') \rangle \parallel \mu\beta.\langle (\kappa', \beta) \parallel \gamma \rangle \right\rangle$$

Et on a donc :

$$d \rightarrow_h \left\langle (\kappa', \tilde{\mu}x.\langle \alpha \parallel (x, \kappa'') \rangle) \parallel \gamma \right\rangle$$

6.3 Séquents sans coupures

On s'intéresse ici au lien entre les preuves dans le calcul des séquents LL original et les dérivations de typage des termes du système L en logique linéaire.

À toute preuve d'un séquent $\vdash A, \Gamma$, on associe la dérivation de type d'une commande ou d'un terme de L, de la forme $c : (\vdash \kappa : A, \Gamma)$ ou $\vdash v : A, \Gamma$. Ce terme ou cette commande est obtenu en suivant à la lettre la structure de la preuve, des

¹¹Terminologie de [DP04].

feuilles à la racine ; durant cette construction, on remarque qu'il est nécessaire pouvoir transformer des commandes en termes et vice-versa, afin de coller à la forme générale des prémisses. À ces endroits-là, on ajoute la règle activation ou la règle désactivation adéquate.

Il peut être nécessaire d'utiliser la convention « atomes neutres » décrite dans la section précédente et donc faire intervenir des « variables neutres » dans la traduction.

Réciproquement, à tout terme ou commande de L accompagné de sa dérivation de type dans LL, on associe la preuve du séquent correspondant en supprimant simplement les règles d'activation, mais aussi celles de désactivation, *i.e.* les coupures dont l'une des prémisses est une règle axiome.

Aux commandes ou termes sans coupures correspondent alors les preuves sans coupures. En effet, on rappelle la définition : une commande est *sans coupure* si ses sous-commandes sont toutes de l'une des formes : $\langle V_+ \parallel \alpha \rangle$, $\langle V_+ \parallel \text{tp} \rangle$ ou $\langle x \parallel V_\ominus \rangle$. Or, le terme ou la commande étant typable dans LL, aucune sous-commande n'est de la forme $\langle V_+ \parallel \text{tp} \rangle$. Les seules occurrences de la règle *cut* sont donc des désactivations ; or celles-ci disparaissent lors de la traduction.

7 La logique classique du second ordre

La syntaxe du système L permet d'accueillir le système LK^{tq} apparu dans [DJS95]. La stratégie de réduction \rightarrow_h caractérise alors ce qui ressemble au sous-système LK_{pol}^η de LK^{tq} . La réduction \rightarrow_r incluse dans \rightarrow_h permet en effet de se passer de l'« η -contrainte ».

On entend dire que \oplus et \wp sont convertibles l'un en l'autre en présence des contractions et affaiblissements, et de même pour \otimes et $\&$, et qu'on en déduit que ces quatre connecteurs n'en forment plus que deux, que l'on note \vee et \wedge , une fois plongés dans la logique « classique ». Pourtant, ces contractions et affaiblissements ne sont pas neutres du point de vue calculatoire.

On a vu que les deux conjonctions et les deux disjonctions se comportent différemment, bien qu'elles soient équivalentes deux-à-deux au niveau de la prouvabilité. On s'intéresse cependant à la procéduralité de l'élimination des coupures

(le niveau -3 de Girard [Gir06]). L'équivalence pour la prouvabilité ne suffit donc pas à identifier de cette sorte deux connecteurs.

En prétendant au contraire qu'ils se différencient au niveau -3 , entre connecteurs paresseux d'une part et connecteurs stricts de l'autre, on permet d'expliquer pourquoi, dans le traitement de la logique classique, il est utile de considérer les quatre connecteurs à la fois, malgré l'absence de contraintes sur les contractions et les affaiblissements.

Le système présenté ici pourra donc être vu comme une version bas-niveau, procédurale, de la logique classique. C'est pourquoi on garde les notations de la logique linéaire pour les connecteurs. Cette distinction ne doit cependant pas nécessairement faire surface au niveau du discours, c'est pourquoi on rejette les notations $\wedge_m, \vee_m, \wedge_a, \vee_a$, auxquelles on préférera, par exemple, l'interprétation à la base du système LC [Gir07], qui avec deux seuls connecteurs \vee et \wedge dénote les quatre autres en fonction de la polarité des sous-formules.

La présentation monolatérale faite ici correspond à celle de Laurent [Lau02], avec une différence majeure : Si l'on souhaite replier le calcul tout en gardant la même expressivité procédurale, il est nécessaire de prendre au sérieux le caractère exponentiel du \neg . On admet donc les exponentielles $!, ?$ dans notre calcul. Une justification possible est que des locutions telles que « *supposons par l'absurde* » ou « *on en déduit par l'absurde* » peuvent nécessiter un sens d'exécution précis.

Le système LK^{tq} est défini fig. 5.

Théorème 3 (Lemme d'adéquation – LK). *Soit c une commande (respectivement v un terme) typable dans LK, de type $\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n$ (resp. $\vdash v : A; \kappa_1 : A_1, \dots, \kappa_n : A_n$) avec A_1, \dots, A_n (resp. et A) des formules closes. Alors pour tous termes clos v_1, \dots, v_n , si $v_1 \Vdash A_1^\perp, \dots, v_n \Vdash A_n^\perp$, alors $c[v_1/\kappa_1, \dots, v_n/\kappa_n] \in \perp$ (resp. $v[v_1/\kappa_1, \dots, v_n/\kappa_n] \Vdash A$).*

Démonstration. Même preuve que pour LL. \square

8 Système L polarisé

La polarisation est une contrainte sur les formules introduite par Laurent dans [Lau02]. Les

formules *polarisées* sont engendrées par la syntaxe suivante :

$$\begin{aligned} P &::= X \mid R \mid P \otimes P \mid P \oplus P \mid \mathbf{1} \mid \mathbf{0} \mid !N \mid \exists X P \\ N &::= X^\perp \mid R^\perp \mid N \wp N \mid N \& N \mid \perp \mid \top \mid ?P \mid \forall X N \end{aligned}$$

La première conséquence est que dorénavant les seuls médiateurs entre la classe positive et la classe négative sont les exponentielles.

8.1 Syntaxe

La polarisation introduit une simplification de la syntaxe des termes : on ne se sert, dans le cadre polarisé, que d'un fragment du système L. De plus, comme vu en remarque 5, on peut se passer du constructeur $\{\cdot\}$ pour dénoter l'extraction.

Voici donc la syntaxe du système L polarisé :

$$\begin{aligned} v_+ &::= x && \mid \mu\alpha.c \\ &&& \mid (v_+, v_+) && \mid \iota_i(v_+) \text{ pour } i \in \{1, 2\} \quad (\otimes, \oplus_i) \\ &&& \mid () && \quad (\mathbf{1}) \\ &&& \mid \mu'(x).c && \quad (!) \\ v_\ominus &::= \alpha && \mid \tilde{\mu}x.c \\ &&& \mid \tilde{\mu}(x, y).c && \mid \tilde{\mu}(\iota_1(x).c \mid \iota_2(y).c) \quad (\wp, \&) \\ &&& \mid \tilde{\mu}().c && \mid \text{tp} \quad (\perp, \top) \\ &&& \mid \wedge(v_+) && \quad (?) \\ c &::= \langle v_+ \parallel v_\ominus \rangle && \mid \langle v_\ominus \parallel v_+ \rangle \end{aligned}$$

On a vu en remarque 5 que l'on peut se passer de marquer l'extraction et la généralisation dans la syntaxe des termes. $|\exists X P|$ et $|\forall X N|$ sont alors redéfinis dans le cadre du système L polarisé par :

$$|\exists X P| = \left(\bigcup \left\{ |P[R/X]| \mid R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V}) \right\} \right)^{\perp\perp}$$

et :

$$|\forall X N| = \bigcap \left\{ |N[R/X]| \mid R \in \mathcal{P}(\mathcal{V}_+^0 \cap \mathbb{V}) \right\}$$

On a déjà vu que ce changement est inoffensif dans le cadre polarisé.

8.2 Déploiements du système L polarisé

Définition 24. Le *déploiement* d'un terme de L est le choix d'une orientation $\langle v_+ \parallel v_\ominus \rangle$ ou $\langle v_\ominus \parallel v_+ \rangle$ — pour chacune des commandes du terme, afin de lui donner un sens « machine abstraite » où le code est par tradition à gauche et

Le groupe identité et le groupe logique sont ceux de la logique linéaire.

Groupe structurel

Règles promotion, déréluction, affaiblissement, contraction :

$$\begin{array}{c}
\frac{\vdash v : A; \Gamma}{\vdash \text{'}(v) : ?A; \Gamma} \text{ (?d)} \quad \frac{c : (\vdash \kappa : A, \Gamma)}{\vdash \mu^{\text{'}}(\kappa).c : !A; \Gamma} \text{ (!)} \\
\\
\frac{c : (\vdash \Gamma)}{c : (\vdash x : A, \Gamma)} \text{ (w)} \quad \frac{c : (\vdash x : A, y : A, \Gamma)}{c[x/y] : (\vdash x : A, \Gamma)} \text{ (c)} \\
\\
\frac{\vdash v : A; \Gamma}{\vdash v : A; x : B, \Gamma} \text{ (w)} \quad \frac{\vdash v : A; x : B, y : B, \Gamma}{\vdash v[x/y] : A; x : B, \Gamma} \text{ (c)}
\end{array}$$

FIG. 5: La logique classique LK^{tq} .

l'environnement à droite.

Afin qu'un tel dépliement ait un sens en informatique, on pourra demander que la condition de réalisme suivante soit remplie : une variable κ n'apparaît jamais à la fois sous la forme $\langle \kappa \rangle$ et sous la forme $|\kappa\rangle$ — c'est l'extension de la distinction entre code et environnement aux variables. On pourra de plus demander que cette propriété soit stable par réduction.

Pour le système L polarisé, deux dépliements triviaux remplissent ces conditions : sa lecture positive et sa lecture négative.

Lecture positive du système L polarisé

Il s'agit du dépliement où toutes les commandes sont de la forme $\langle v_+ \parallel v_\ominus \rangle$, donc où l'on dispose de la conjonction et de la disjonction strictes.

Posons $\neg P = !P^\perp$. On écrit les dérivations de type de façon bilatérale, où les négatifs sont à gauche et les positifs à droite ; cela donne des séquents de la forme $P_1, \dots, P_n \vdash Q_1, \dots, Q_m$, où toutes les formules sont écrites de façon positive. Ces formules positives sont définies de la façon suivante :

$$P, Q ::= X \mid P \otimes Q \mid P \oplus Q \mid 1 \mid 0 \mid \exists X P \mid \neg P$$

De plus, on écrira v pour v_+ et e (comme environnement) pour v_\ominus , à la façon de [CH00].

Définition 25 (Implication en appel par valeur). On pose $P \rightarrow Q = \neg(P \otimes \neg Q)$ et :

$$\begin{array}{l}
\lambda x.v = \mu^{\text{'}}(x, y). \langle y \parallel \text{'}(v) \rangle \quad (y \notin \mathcal{FV}(v)) \\
v \cdot e = \text{'}(v, \mu^{\text{'}}(x). \langle x \parallel e \rangle) \quad (x \notin \mathcal{FV}(e)) \\
vv' = \mu\alpha. \langle v \parallel v' \cdot \alpha \rangle \quad (\alpha \notin \mathcal{FV}(v, v'))
\end{array}$$

Proposition 14. On a les dérivations suivantes :

$$\begin{array}{c}
\frac{\Gamma, x : P \vdash v : Q; \Delta}{\Gamma \vdash \lambda x.v : P \rightarrow Q; \Delta} (\rightarrow) \\
\\
\frac{\Gamma \vdash v : P; \Delta \quad \Gamma'; e : Q \vdash \Delta'}{\Gamma, \Gamma'; v \cdot e : P \rightarrow Q \vdash \Delta, \Delta'} (\rightarrow \vdash) \\
\\
\frac{\Gamma \vdash v : P \rightarrow Q; \Delta \quad \Gamma' \vdash v' : P; \Delta'}{\Gamma, \Gamma' \vdash vv' : Q; \Delta, \Delta'}
\end{array}$$

On a les règles de réduction suivantes :

$$\begin{array}{l}
\langle vv' \parallel e \rangle \rightarrow_h \langle v \parallel v' \cdot e \rangle \\
\langle \lambda x.v \parallel v' \cdot e \rangle \rightarrow^* \langle v' \parallel \tilde{\mu}x. \langle v \parallel e_\eta \rangle \rangle \quad (x \notin \mathcal{FV}(e))
\end{array}$$

où $e_\eta = \tilde{\mu}x. \langle x \parallel e \rangle$.

Sachant que l'on a déjà la règle $\langle V \parallel \tilde{\mu}x. \langle v \parallel e \rangle \rangle \rightarrow \langle v[V/x] \parallel e \rangle$ lorsque $x \notin \mathcal{FV}(e)$, notre implication correspond donc essentiellement à celle du calcul $\tilde{\lambda}\tilde{\mu}$ en appel par valeur défini dans [CH00].

Définition 26 (Polymorphisme en appel par valeur). On pose $\forall X P = \neg(\exists X \neg P)$ et :

$$\begin{aligned}\Lambda X.v &= \mu^{\lambda}(x).\langle x \parallel \lambda(v) \rangle & (x \notin \mathcal{FV}(v)) \\ P\{e\} &= \lambda^{\mu}(x).\langle x \parallel e \rangle & (x \notin \mathcal{FV}(e)) \\ \{v\}P &= \mu\alpha.\langle v \parallel P\{\alpha\} \rangle & (\alpha \notin \mathcal{FV}(v))\end{aligned}$$

Les notations sont celles employées dans [Gir06] — le fait d'écrire des formules dans les termes n'a que peu d'importance ; ici, cela permet plus de clarté. La proposition qui suit signifie que l'on a ainsi implémenté le polymorphisme en appel par valeur, qui vérifie en particulier le fait que $\Lambda X.v$ est une valeur.

Proposition 15. On a les dérivations suivantes :

$$\frac{\Gamma \vdash v : P; \Delta}{\Gamma \vdash \Lambda X.v : \forall X P; \Delta} (\vdash \forall) \quad (X \notin \mathcal{FV}(\Gamma, \Delta))$$

$$\frac{\Gamma; e : P[Q/X] \vdash \Delta}{\Gamma; Q\{e\} : \exists X P \vdash \Delta} (\forall \vdash)$$

$$\frac{\Gamma \vdash v : \forall X P; \Delta}{\Gamma \vdash \{v\}Q : P[Q/X]; \Delta}$$

On a les règles de réduction suivantes :

$$\begin{aligned}\langle \{v\}P \parallel e \rangle &\rightarrow_h \langle v \parallel P\{e\} \rangle \\ \langle \Lambda X.v \parallel P\{e\} \rangle &\rightarrow_h^* \langle v \parallel e_\eta \rangle \quad (x \notin \mathcal{FV}(e))\end{aligned}$$

où $e_\eta = \tilde{\mu}x.\langle x \parallel e \rangle$.

Lecture négative du système L polarisé

Il s'agit du dépliement où toutes les commandes sont de la forme $\langle v_\ominus \parallel v_+ \rangle$, donc où l'on dispose de la conjonction et de la disjonction passeuses.

On pose cette fois-ci $\neg N \equiv ?N^\perp$. Les positifs sont maintenant à gauche du tourniquet et les négatifs sont à droite de celui-ci. Les jugements sont donc de la forme $N_1, \dots, N_n \vdash M_1, \dots, M_m$. Ces formules négatives sont écrites de la façon suivante :

$$N, M ::= X \mid N \& M \mid N \wp M \mid \top \mid \perp \mid \forall X N \mid \neg N$$

On écrit v les v_\ominus et e les v_+ .

Lorsqu'on a écrit la syntaxe du système L, on a repris pour les termes positifs la notation des

termes du calcul de Curien-Herbelin (ce qu'ici nous nommons code), et pour les termes négatifs la notation de leurs contextes (ce qu'ici nous nommons environnement). Si cela assure la compatibilité des notations avec celles des travaux de Curien-Herbelin pour la lecture positive, pour la lecture négative nous sommes à contre-sens. Il faudra en effet lire α comme une variable et x comme une co-variable ; de même, $\mu\alpha$ se lira « let $\alpha = []$ in » tandis que $\tilde{\mu}x$ capturera les environnements. Que le lecteur pardonne les notations de cette section !

Définition 27 (Implication en appel par nom). On pose $N \rightarrow M = \neg N \wp M$ et :

$$\begin{aligned}\lambda\alpha.v &= \tilde{\mu}(x, y).\langle \lambda^{\mu}(\mu\alpha.\langle v \parallel y \rangle) \parallel x \rangle & (x, y \notin \mathcal{FV}(v)) \\ v \cdot e &= (\mu^{\lambda}(x).\langle v \parallel x \rangle, e) & (x \notin \mathcal{FV}(v)) \\ vv' &= \tilde{\mu}x.\langle v \parallel v' \cdot x \rangle & (x \notin \mathcal{FV}(v, v'))\end{aligned}$$

Proposition 16. On a les dérivations suivantes :

$$\frac{\Gamma, \alpha : N \vdash v : M; \Delta}{\Gamma \vdash \lambda\alpha.v : N \rightarrow M; \Delta} (\vdash \rightarrow)$$

$$\frac{\Gamma \vdash v : N; \Delta \quad \Gamma'; e : M \vdash \Delta'}{\Gamma, \Gamma'; v \cdot e : N \rightarrow M \vdash \Delta, \Delta'} (\rightarrow \vdash)$$

$$\frac{\Gamma \vdash v : N \rightarrow M; \Delta \quad \Gamma' \vdash v' : N; \Delta'}{\Gamma, \Gamma' \vdash vv' : M; \Delta, \Delta'}$$

On a les règles de réduction suivantes (où E est une valeur) :

$$\begin{aligned}\langle vv' \parallel E \rangle &\rightarrow_h^* \langle v \parallel v' \cdot E \rangle & (\text{push}) \\ \langle \lambda\alpha.v \parallel v' \cdot E \rangle &\rightarrow_h^* \langle v[v'_\eta/\alpha] \parallel E \rangle & (\text{pop})\end{aligned}$$

avec $v'_\eta = \tilde{\mu}x.\langle x \parallel v' \rangle$ ($x \notin \mathcal{FV}(v')$).

Cette implication correspond donc essentiellement à celle des machines de Krivine, ou encore à celle du calcul par nom de Curien-Herbelin. Ce qui suit achève de définir les machines de Krivine de [Kri04] dans L.

Définition 28. On pose pour tout e terme positif, $k_e = \lambda\alpha.\tilde{\mu}_-.\langle \alpha \parallel e \rangle$ pour $\alpha \notin \mathcal{FV}(e)$, ainsi que $cc = \lambda\alpha.\tilde{\mu}x.\langle \alpha \parallel k_x \cdot x \rangle$.

Proposition 17. On a les dérivations suivantes :

$$\frac{\Gamma; e : N \vdash \Delta}{\Gamma \vdash k_e : N \rightarrow M; \Delta}$$

$$\overline{\vdash_{cc} : (((N \rightarrow M) \rightarrow N) \rightarrow N);}$$

On a les règles de réduction suivantes, pour E et E' des valeurs :

$$\begin{aligned} \langle cc \parallel t \cdot E \rangle &\rightarrow_h^* \langle t \parallel k_E \cdot E \rangle \quad (\text{store}) \\ \langle k_E \parallel t \cdot E' \rangle &\rightarrow_h^* \langle t \parallel E \rangle \quad (\text{restore}) \end{aligned}$$

Remarque 8. La réalisabilité définie dans cet article est une généralisation de la réalisabilité définie dans [Kri04]. Remarquer en particulier que la définition de $|\forall X N|$ de [Kri04] n'est pas plus générique que la nôtre contrairement à ce que l'on pourrait craindre, puisque l'auteur restreint les paramètres à des ensembles de *pires*¹², lesquelles sont bien des valeurs.

Le système L peut donc être utile à la poursuite de l'étude du paradigme de la *programmation par preuves* [Kri90, Kri93]. Observer par exemple que la définition du type des booléens $\text{Bool}[x] \equiv (x = 0) \oplus (x = 1)$ dans L est plus forte que celle du système $F : \forall X (X 1 \rightarrow X 0 \rightarrow X x)$. En effet, on obtient directement le principe équivalent mais plus générique : $\forall X ((x = 1 \rightarrow X 1) \rightarrow (x = 0 \rightarrow X 0) \rightarrow X x)$.

9 La logique linéaire polarisée du second ordre

Le système L polarisé est un quotient de la logique linéaire polarisée telle que celle-ci fut introduite par Laurent [Lau02]. Linéaire signifie ici linéaire sur les positifs ; comme il y a au plus un positif dans un séquent, il n'est nécessaire d'avoir qu'une seule variable négative distinguée, que l'on choisit de noter \star .

9.1 Dérivations

Définition 29. \mathcal{N} désigne un contexte constitué de formules négatives uniquement et Γ désigne un contexte de la forme \mathcal{N} ou $\mathcal{N} \cup \{\star : P\}$. Les séquents de LLP sont des jugements de la forme suivante :

$$\begin{aligned} c : & (\vdash \Gamma) \\ & \vdash v_+ : P; \mathcal{N} \\ & \vdash v_\ominus : N; \Gamma \end{aligned}$$

¹²Les environnements $v_1 \cdot v_2 \cdots v_n \cdot \pi_0$ avec $v_1, \dots, v_n \in \mathcal{V}_\ominus^0$ et π_0 une constante.

La formule positive, lorsqu'elle existe, est dite *focalisée*. Si l'on veut respecter les conventions, on l'écrira tout à droite du contexte, avec une exception dans le cas où elle est activée.

La logique linéaire polarisée du second ordre est définie fig. 6.

9.2 Lemme d'adéquation

Théorème 4 (Lemme d'adéquation – LLP). *Soit c une commande (respectivement v un terme) typable dans LLP, de type $\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n$ (resp. $\vdash v : A; \kappa_1 : A_1, \dots, \kappa_n : A_n$) avec A_1, \dots, A_n (resp. et A) des formules closes. Alors pour tous termes clos v_1, \dots, v_n si $v_1 \Vdash A_1^\perp, \dots, v_n \Vdash A_n^\perp$, alors $c[v_1/\kappa_1, \dots, v_n/\kappa_n] \in \perp$ (resp. $v[v_1/\kappa_1, \dots, v_n/\kappa_n] \Vdash A$).*

Démonstration. Preuve similaire à celle pour LL. \square

Conclusion

La théorie du calcul actuelle repose sur le λ -calcul, lequel possède une caractérisation peu naturelle de l'appel par valeur. L'appel par nom est, quant à lui, la stratégie de réduction « *envoyée par dieu* » et donc la préférée des théoriciens, tandis que si la plupart des langages implémentent l'appel par valeur, c'est par *pragmatisme*, car il permet de rendre compte des *effets de bord*.

La lignée des travaux dans lequel se situe le présent, qui a commencé par celui, précurseur, de Filinski [Fil89] et s'est poursuivie par ceux de Selinger [Sel01] et de Curien-Herbelin [CH00], fait apparaître cette conception des stratégies de réduction, qui suit une dichotomie théorique-pratique, comme un artifice.

Contrairement au λ -calcul, le système L est impliqué dans les stratégies de réduction, puisque l'appel par nom et l'appel par valeur sont traités de façon uniforme et non *ad hoc*. En outre, alors que le λ -calcul proposait la correspondance *preuve-programme*, L propose la correspondance *calcul des séquents-machine abstraite* et en ce sens va plus loin. En effet, une machine abstraite est un cadre interactif dans lequel on peut redéfinir¹³, entre autres, les λ -calculs et les langages

¹³*Effet diligence* : le nouveau commence par imiter l'ancien. (Jacques Perriault : « *Les premiers wagons avaient la forme des diligences* ».)

Groupe identité

Règles axiome, activation, coupure :

$$\frac{}{\vdash x : P; x : P^\perp} \text{(ax+)} \quad \frac{}{\vdash \star : N; \star : N^\perp} \text{(ax}\ominus\text{)}$$

$$\frac{c : (\vdash x : N, \Gamma)}{\vdash \tilde{\mu}x.c : N; \Gamma} \text{(\tilde{\mu})} \quad \frac{c : (\vdash \Gamma, \star : P)}{\vdash \mu\star.c : P; \Gamma} \text{(\mu)}$$

$$\frac{\vdash v : A; \Gamma \quad \vdash v' : A^\perp; \Delta}{\langle v \parallel v' \rangle : (\vdash \Gamma, \Delta)} \text{(cut)}$$

Groupe structurel

Règles promotion, dérélliction, affaiblissement, contraction :

$$\frac{\vdash v : P; \Gamma}{\vdash \lambda(v) : ?P; \Gamma} \text{(?d)} \quad \frac{c : (\vdash x : N, \mathcal{N})}{\vdash \mu^\lambda(x).c : !N; \mathcal{N}} \text{(!)}$$

$$\frac{c : (\vdash \Gamma)}{c : (\vdash x : N, \Gamma)} \text{(?w)} \quad \frac{c : (\vdash x : N, y : N, \Gamma)}{c[x/y] : (\vdash x : N, \Gamma)} \text{(?c)}$$

$$\frac{\vdash v : A; \Gamma}{\vdash v : A; x : N, \Gamma} \text{(?w)} \quad \frac{\vdash v : A; x : N, y : N, \Gamma}{\vdash v[x/y] : A; x : N, \Gamma} \text{(?c)}$$

Groupe logique

Règles tenseur, par, avec, plus :

$$\frac{\vdash v : P; \Gamma \quad \vdash v' : Q; \Delta}{\vdash (v, v') : P \otimes Q; \Gamma, \Delta} \text{(\otimes)} \quad \frac{c : (\vdash x : N, y : M, \Gamma)}{\vdash \tilde{\mu}(x, y).c : N \wp M; \Gamma} \text{(\wp)}$$

$$\frac{c : (\vdash x : N, \Gamma) \quad c' : (\vdash y : M, \Gamma)}{\vdash \tilde{\mu}(\iota_1(x).c \mid \iota_2(y).c') : N \& M; \Gamma} \text{(\&)} \quad \frac{\vdash v : P_i; \Gamma}{\vdash \iota_i(v) : P_1 \oplus P_2; \Gamma} \text{(\oplus_{i \in \{1,2\}})}$$

Extraction, généralisation :

$$\frac{\vdash v : P[Q/X]; \Gamma}{\vdash v : \exists X P; \Gamma} \text{(\exists)} \quad \frac{\vdash v : N, \Gamma}{\vdash v : \forall X N; \Gamma} \text{(\forall) } (X \notin \mathcal{FV}(\Gamma))$$

Règles unités :

$$\frac{}{\vdash () : \mathbf{1}; \Gamma} \text{(1)} \quad \frac{c : (\vdash \Gamma)}{\vdash \tilde{\mu}().c : \perp; \Gamma} \perp$$

$$\frac{}{\vdash \top : \overline{\top}; \Gamma} \text{(\top)} \quad \text{pas de règle pour } \mathbf{0}$$

FIG. 6: La logique linéaire polarisée du second ordre

de programmation usuels. Cela donne en outre un cadre théorique pour justifier et prolonger des travaux qui se placent dans un cadre de machines abstraites afin de bénéficier d'un contrôle sur l'évaluation, comme les travaux de Krivine [Kri04].

Enfin, on espère avoir contribué à montrer que la théorie moderne de la démonstration, dans ses développements récents, s'est rapproché de l'informatique plutôt qu'elle ne s'en est éloignée, puisque si nous retrouvons les notions de machine abstraite, d'appel par valeur et d'appel par nom, ce n'est pas par des considérations *ad hoc*, mais par l'application des notions de la théorie de la démonstration que sont la polarisation et la réversibilité des négatifs.

Remerciements

Mes remerciements vont à Hugo Herbelin, tout d'abord, en raison de nombreuses discussions et commentaires sur ce travail. Je remercie également Pierre-Louis Curien, Stephan Zdancewic ainsi que Jeffrey Vaughan.

Références

- [AH03] Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *Thirtieth International Colloquium on Automata, Languages and Programming, ICALP '03, Eindhoven, The Netherlands, June 30 - July 4, 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 871–885. Springer, 2003.
- [AHS04] Zena M. Ariola, Hugo Herbelin, and Amr Sabry. A type-theoretic foundation of continuations and prompts. In *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming (ICFP '04), Snowbird, Utah, September 19-21, 2004*, pages 40–53. ACM, 2004.
- [BTKP93] Val Breazu-Tannen, Delia Kesner, and Laurence Puel. A typed pattern calculus. In *Eighth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 262–274, Los Alamitos, CA, June 1993. IEEE Computer Society Press.
- [CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. *ACM SIGPLAN Notices*, 35(9) :233–243, 2000.
- [Coe05] Claudio Sacerdoti Coen. Explanation in natural language of $\lambda\mu\tilde{\mu}$ -terms. In *MKM*, pages 234–249, 2005.
- [DJS95] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A new deconstructive logic : Linear logic. 1995.
- [DP04] Kosta Dosen and Zoran Petric. *Proof-theoretical Coherence (Studies in Logic S.)*. King's College Publications, December 2004.
- [Fil89] Andrzej Filinski. Declarative continuations and categorical duality. Master's thesis, 1989.
- [FT04] Carsten Führmann and Hayo Thielecke. On the call-by-value CPS transform and its semantics. *Information and Computation*, 188(2) :241–283, 2004.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [Gir06] Jean-Yves Girard. *Le Point Aveugle, Cours de logique, Tome I : Vers la Perfection*. Vision des Sciences. Hermann, 2006.
- [Gir07] Jean-Yves Girard. *Le Point Aveugle, Cours de logique, Tome II : Vers l'imperfection*. Visions des Sciences. Hermann, 2007.

[AH03, CH00, Coe05, GTL89, Gir06, Gir07, Sel01, Sel03, Wad03, Her05, Lau02, Gir87, Kri93]

- [GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University Press, 1989. 165 p.
- [Her05] Hugo Herbelin. C'est maintenant qu'on calcule, au cœur de la dualité. <http://yquem.inria.fr/~herbelin/habilitation/index.html>, 2005. Habilitation thesis.
- [Her08] Hugo Herbelin. Duality of computation and sequent calculus : a few more remarks. Manuscript, 2008.
- [Kri90] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Masson, 1990. 175 p.
- [Kri93] Jean-Louis Krivine. *Lambda-calculus, types and models*. Ellis Horwood, 1993. Available on the Internet.
- [Kri04] Jean-Louis Krivine. Realizability in classical logic. To appear in *Panoramas et synthèses*, Société Mathématique de France., 2004.
- [Lau02] Olivier Laurent. *Etude de la polarisation en logique*. Thèse de doctorat, Université Aix-Marseille II, mar 2002.
- [Sel01] Peter Selinger. Control categories and duality : On the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11(2) :207–260, 2001.
- [Sel03] Peter Selinger. Some remarks on control categories. Manuscript, 2003.
- [Wad03] Philip Wadler. Call-by-value is dual to call-by-name. *SIGPLAN Not.*, 38(9) :189–201, 2003.
- [Wad04] Philip Wadler. Down with the bureaucracy of syntax! pattern matching for classical linear logic. Unpublished draft. Google : wadler+dual, 2004.