Etude polarisée du système L

Guillaume Munch-Maccagnoni

Étude polarisée du système L

Guillaume MUNCH–MACCAGNONI*
INRIA Saclay Univ. of Pennsylvania

L'article s'intéresse au *système* L, nom qu'Herbelin a proposé pour faire référence à une syntaxe générique pour l'étude des calculs des séquents, qui hérite du calcul $\bar{\lambda}\mu\tilde{\mu}$ de Curien et Herbelin.

On munit le système L d'un protocole de réduction *focalisant* basé sur la notion de polarité, héritée du calcul des séquents LC, la première et meilleure version constructive de LK, et on en donne une lecture informatique précise.

On obtient un même quotient littéral pour les calculs des séquents majeurs (LK, LL...) à la fois dans la tradition de Gentzen et dans celle de Girard. En particulier, le quotient de LC répond à une question de Girard. La focalisation et le *pattern matching* apparaissent de façon non bureaucratique. La dualité du calcul est reformulée.

Le système L focalisant est un cadre naturel pour l'expression de la *réalisabilité classique* de Krivine. On s'en inspire pour le munir d'une réalisabilité conforme en outre à l'esprit des *comportements* de la Ludique de Girard. La réalisabilité à la Krivine se trouve alors étendue, en particulier à l'appel par valeur.

Comme application, le système et l'outil offrent un cadre théorique pour l'étude de la programmation typée qui offre un bon traitement des notions d'ordre d'évaluation et d'effets de bord, étendant ainsi le travail de Zeilberger, qui montre le besoin et la possibilité d'un tel cadre, à une configuration conventionnelle et plus puissante. La méthode permet d'analyser les définitions possibles du polymorphisme en présence d'effets de bord, avec un lien direct avec les « égalités choquantes » de la Ludique.

*Janvier 2009.

Table des matières

1 Introduction

Lorsque Curien et Herbelin mettent au jour dans [CH00] la structure calculatoire des séquents, ils exhibent un modèle de calcul mettant en œuvre une interaction simple entre *code* v et *environnement* e au sein de *commandes* $\langle v \parallel e\rangle$. C'est ce que l'on appelait le calcul $\bar{\lambda}\mu\tilde{\mu}$, et que l'on appellera, en suivant la suggestion d'Herbelin [Her08][1], le système L, par référence à cette tradition de donner aux calculs des séquents des noms commençant par cette lettre.

Ce modèle est très pur du point de vue informatique, puisqu'il rappelle la vision traditionnelle du calcul comme l'interaction d'un *programme* avec une *donnée,* que l'on trouve déjà aux origines de la science, avec la théorie des automates.

L'analogie n'est pas vaine. Proposons à cette fin d'écrire $\langle \omega \parallel s\rangle$ pour symboliser l'interaction d'un mot $\omega \in \Sigma^*$ avec un automate $\mathcal{A} = (S, \Sigma, R, s_0, S_F)$ dans un état $s \in S$, par analogie avec les commandes $\langle v \parallel e\rangle$ du système L que l'on retrouvera plus loin. On définit alors la relation de « réduction » $\rightarrow$ de cet automate par $\langle a.\omega \parallel s\rangle \rightarrow \langle \omega \parallel s'\rangle$ lorsque $(s, a, s') \in R$ est une transition de l'automate.

1.1 « Réalisabilité » pour automates finis

Afin d'introduire l'outil de *réalisabilité classique* qui sera développée ici pour le système L focalisant, poursuivons l'analogie en définissant une notion similaire sur notre automate.

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

On définit une notion d'orthogonalité entre mots de Σ^* et états de S en se donnant une *observation* $\perp\!\!\!\perp$ qui est une partie distinguée de l'ensemble des $\langle \omega \parallel s \rangle$ vérifiant la propriété d'être *saturée*, c'est-à-dire qui vérifie pour tous $\omega, \omega' \in \Sigma^*$ et $s, s' \in S$:

$$\langle \omega \parallel s \rangle \to \langle \omega' \parallel s' \rangle \in \perp\!\!\!\perp \Longrightarrow \langle \omega \parallel s \rangle \in \perp\!\!\!\perp$$

On note alors $\omega \perp\!\!\!\perp s$ lorsque $\langle \omega \parallel s \rangle \in \perp\!\!\!\perp$.

Pour une observation $\perp\!\!\!\perp$ donnée, on définit alors :

- Pour tout $\mathcal{L} \subset \Sigma^*$:

$$\mathcal{L}^\perp = \{\, s \in S \mid \forall \omega \in \mathcal{L}, \omega \perp\!\!\!\perp s \,\}$$

- Pour tout $\mathcal{S} \subset S$:

$$\mathcal{S}^\perp = \{\, \omega \in \Sigma^* \mid \forall s \in \mathcal{S}, \omega \perp\!\!\!\perp s \,\}$$

Les ensembles de la forme $\mathcal{L}^\perp$ ou $\mathcal{S}^\perp$ ne sont pas quelconques. Par exemple, intéressons-nous au cas où l'on pose $\perp\!\!\!\perp$ la plus petite observation pour laquelle $\varepsilon \perp\!\!\!\perp s_F$ pour tout $s_F \in S_F$ (un choix naturel) ; il s'agit de :

$$\perp\!\!\!\perp = \left\{ \langle \omega \parallel s \rangle \;\middle|\; \exists s_F \in S_F, \langle \omega \parallel s \rangle \to^* \langle \varepsilon \parallel s_F \rangle \right\}$$

D'une part, les $\mathcal{S}^\perp$ sont des langages rationnels, avec en particulier $\{s_0\}^\perp$ le langage reconnu par $\mathcal{A}$. Ceux-ci sont décrits par les expressions rationnelles : on montrera par exemple que pour tel automate, $\{s_0\}^\perp = |a^*(ba)b^*|$, lorsque $|A|$ désigne le langage décrit par l'expression régulière A. D'autre part, la colinéarité de s et de s', c'est-à-dire $\{s\}^\perp = \{s'\}^\perp$, correspond à l'équivalence de Nérode des états s et s'. La classe d'équivalence de s est alors donnée par $\{s\}^{\perp\perp}$.

On a ainsi exprimé dans un formalisme succinct deux des axes principaux de la théorie des automates.

1.2 Système L et comportements

Plutôt que de mots et d'automates, les structures dont il est question ici sont de nature logique, comme on en rencontre en programmation fonctionnelle. Il s'agit de preuves/programmes construits à partir des constructeurs issus de la logique linéaire : la paire (t, u) pour le tenseur $\otimes$; le pattern-matching sur cette paire $\mu(x, y).c$ pour le par $\parr$; l'injection $\iota_i(t)$ pour le plus $\oplus$; et le pattern-matching correspondant $\mu(\iota_1(x).c \,|\, \iota_2(y).c')$ pour l'avec &.

Les ensembles de termes de la forme $U^\perp$ sont ce que l'on conviendra d'appeler des *comportements*. Au lieu d'expressions rationnelles, ce sont les formules de la logique qui décriront certains comportements du système L. Une question que l'on se posera sans cesse sera : étant donné un terme de preuve t et une formule logique A, le premier appartient-il au comportement décrit par la seconde ? (ce que l'on écrit $t \Vdash A$ et prononce « t réalise A »).

Cette manière de voir l'activité logique, empruntée à la Ludique de Girard [Gir07], permet ainsi de fonder après-coup le domaine de la programmation par preuves [Kri90, Kri93] ou l'étude des systèmes de types pour les langages de programmation. En effet, les systèmes de preuve peuvent permettre de répondre à cette question, à savoir que si l'on parvient à donner à t le type A, alors t réalise A quelle que soit l'observation $\perp\!\!\!\perp$.

Les systèmes de preuves dont il est question ici sont, à travers une syntaxe asymétrique unique, la logique linéaire LL et une variante calculatoire de la logique classique LK que l'on introduit ($\mathsf{LK}_{\mathrm{pol}}$), à la fois dans la tradition bilatère (Gentzen) et celle monolatère (Girard).

1.3 Système L et protocole de réduction focalisant

La question de la stratégie de réduction est centrale pour les interprétations calculatoires du calcul des séquents. Tandis que dans le λ-calcul, la propriété de Church-Rosser permet de reléguer cette question au second plan, ce n'est pas le cas du calcul des séquents, qui comporte un certain nombre de paires critiques : si l'on est $\langle \mu\_.c \parallel \mu\_.c' \rangle$ — la *paire critique de Lafont* —, doit-on se réduire en c ou bien en c' ? Si l'on est $\langle (v, v') \parallel \mu(x, y).c \rangle$, comment concilier les deux possibilités, $\left\langle v \,\middle\|\, \mu x.\langle v' \parallel \mu y.c \rangle \right\rangle$ et $\left\langle v' \,\middle\|\, \mu y.\langle v \parallel \mu x.c \rangle \right\rangle$?

La conception des stratégies de réduction qui prévaut dans les travaux sur la « *dualité du calcul* » : le calcul $\bar{\lambda}\mu\tilde{\mu}$ de Curien et Herbelin [CH00, Her05, Her08] ou le calcul dual de Wadler [Wad03], est que la stratégie de réduction ou bien donne au code le contrôle sur l'environnement — on a alors $\langle \mu\_.c \parallel \mu\_.c' \rangle \to c$ —, ou bien

donne à l'environnement le contrôle sur le code — on a alors $\langle \mu\_.c \parallel \mu\_.c' \rangle \to c'$ —. Le premier cas permet d'implémenter l'*appel par valeur*, le second permet d'implémenter l'*appel par nom*.

Pourtant, on peut concevoir une multitude de stratégies intermédiaires entre ces deux extrêmes [DJS95], et parmi les possibilités se distingue une stratégie basée sur la polarité, un argument issu des travaux sur la focalisation et développé par Girard dans [Gir91]. C'est cette stratégie *focalisante* que l'on développe ici.

L'appel par nom et l'appel par valeur s'implémentent alors dans des sous-systèmes de L focalisant, ce qui permet de remettre en cause cette conception des stratégies de réduction, qui veut que l'appel par nom et l'appel par valeur émergent d'un *choix* arbitraire entre donner le contrôle au code ou le donner à l'environnement.

1.4 Héritages et connexions

En effet, loin de s'inscrire dans cette dichotomie, ce travail a pour fil conducteur la thèse suivante :

> Les notions informatiques de « *paresse* » et d'« *avidité* »[2] correspondent respectivement aux notions de « *négatif* » et de « *positif* » de la théorie de la démonstration.

Si cette correspondance est acceptable dans l'ordre de l'intuitif, puisqu'il est admis que positif-négatif renvoie à l'opposition actif-passif, il nous faudra cependant soutenir que « *paresseux* » et « *strict* » ne sont pas des qualités de stratégies d'évaluation, mais les qualités des connecteurs de la logique.

Dans la littérature, cette lecture se retrouve explicitement dans le travail de Zeilberger [Zei08a], mais c'est de façon indépendante que ce travail et cette observation ont été réalisés.

Les systèmes polarisés L'introduction des polarités positive et négative en logique est due à Girard[3], qui proposa dans [Gir91] une « logique classique constructive », LC, le but étant de résoudre les paires critiques telles celle de Lafont, ce qui est fait en donnant la priorité dans la coupure au membre positif.

Ce sont les mêmes considérations qui ont donné lieu au $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ de Danos, Joinet et Schellinx [DJS95], qui héritait LC et auquel le système $\mathsf{LK}_{\mathrm{pol}}$ présenté ici est similaire. Outre le fait que ces travaux ne donnaient pas de quotient tractable des systèmes logiques qu'ils proposaient, la différence majeure est que notre configuration nous dispense des entraves syntaxiques nécessaires à la définition de ceux-ci (bénitiers, « η-restriction »).

La dualité du calcul En 1999, Selinger [Sel01] introduit les *catégories de contrôle* ainsi que leurs duales, les *catégories de co-contrôle*, et montre qu'elles admettent chacune pour langage interne un calcul de continuations $\lambda\mu$, en appel par nom pour les premières et en appel par valeur pour les secondes. Il montre un résultat de dualité syntaxique entre ces deux modes d'évaluation, que l'on nomme la dualité du calcul (découverte initialement par Filinski [Fil89]).

En 2000, Curien et Herbelin [CH00] donnent un formalisme ($\bar{\lambda}\mu\tilde{\mu}$) qui donne à l'*environnement* du calcul un statut symétrique à celui des termes. Ils exhibent ainsi cette dualité dans la syntaxe, puisque la différence entre appel par valeur et appel par nom apparaît comme le résultat d'un choix symétrique au niveau de la stratégie de réduction entre (respectivement) donner la priorité au terme ou la donner à l'environnement.

En 2002, Laurent [Lau02] donne comme sémantique à sa logique linéaire polarisée (LLP) les catégories de Selinger : les formules négatives deviennent des objets des catégories de contrôle, et les formules positives des objets des catégories de co-contrôle. Il a alors suggéré ([Lau02], section « Dualité du calcul ») l'étude de la dualité du calcul à travers la notion de polarisation.

L'article hérite directement du $\bar{\lambda}\mu\tilde{\mu}$ et de ses diverses extensions (Wadler [Wad03], Herbelin [Her05]) pour suivre cette voie.

Les transformations CPS Il est connu (Plotkin [Plo75]) que la notion de continuation donne un cadre uniforme à l'étude des stratégies de réduction, à travers les transformations par passage de continuation (CPS). À la fois les systèmes polarisés [Gir91, Lau02] et le $\bar{\lambda}\mu\tilde{\mu}$ [CH00], dont il est

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

[3] La notion est apparue lorsque l'on a dit que A est en position négative dans $A \to B$; mais alors elle n'était pas centrale en logique.

question ici, ont été dépeints comme des façons de décrire des transformations CPS.

À titre d'exemple, Sabry et Felleisen, en étudiant l'image de la transformation CPS en appel par valeur, déduisent de nouvelles équivalences de termes [SF93]. C'est de façon avantageusement directe que l'on obtient ces équivalences dans le calcul de Curien-Herbelin.

De même, on pourra être tenté de restreindre sa vision du système L à celle d'une écriture particulière du λ-calcul en régime CPS. Mais il faudra alors convenir qu'il s'agit là d'une façon bien *lisible* de décrire ces transformations, aussi bien au niveau structurel (grâce à l'héritage du $\bar{\lambda}\mu\tilde{\mu}$) qu'au niveau logique (grâce à l'héritage de LC), d'autant plus qu'à chacun des connecteurs en présence, on parvient à donner un sens informatique.

Les machines abstraites Le travail de Curien et Herbelin établit en outre un lien entre le calcul des séquents et les machines abstraites telles la machine de Krivine — ces formalismes qui décrivent le calcul par l'interaction d'un code et d'un environnement. Outre son lien avec la symétrie du calcul des séquents, voici deux raisons pour lesquelles cette présentation se distingue :

1. Cette présentation apparaît de façon implicite dans la Ludique de Girard. Pour celle-ci, la finalité d'une preuve de A est d'être testée contre une preuve de $A^{\perp}$. Bien sûr, sous peine d'incohérence, cela veut dire que l'une des deux preuves est une fausse preuve : une *para-preuve*. D'une part, il s'agit d'un cadre dans lequel les preuves répondent à des objets de même nature qu'elles : c'est le thème de la dualité moniste [Gir07], et dont le pendant informatique est justement cette représentation du calcul donnant un statut logique à l'environnement. D'autre part, cela donne un statut aux « fausses » preuves, ce que l'on retrouve ici dans l'importance accordée à la syntaxe non typée — en particulier, une commande close ne peut être bien typée, car alors il s'agirait d'une preuve du séquent vide.

   Le point de vue de la Ludique est celui adopté ici. On en divergera cependant sur un point : celui du choix d'un cadre *spiritualiste* plutôt que *locatif*. En se gardant de remises en cause aussi radicales, notre système L a l'ambition de sa modestie puisqu'il gagne en proximité avec la théorie des langages existante, comme on pourra le voir lorsqu'il s'agira de redéfinir les λ-calculs dans notre système.

2. Dans le programme de Krivine [Kri04], celui de donner le contenu calculatoire des axiomes des mathématiques courantes, l'utilisation des machines de Krivine est nécessaire d'une part car celles-ci donnent un contrôle fin sur le calcul, puisque celui-ci s'effectue en réduction de tête, et d'autre part car elles permettent la définition de la réalisabilité classique.

La réalisabilité à la Krivine En raison de cette présentation sous forme de machine abstraite, le système L focalisant fournit un cadre naturel pour l'expression de la réalisabilité classique. Cet outil élégant, qui montre des propriétés de normalisation, de paramétricité, de *type safety*, ou plus généralement de conformation à une spécification que le type décrit, se trouve ici étendu à un système avec une stratégie polarisée, et donc *a fortiori* à l'appel par valeur.

La Ludique met de l'ordre parmi les parapreuves en distinguant des comportements. On s'en inspire pour définir des comportements en termes de réalisabilité classique.

Applications à l'étude des langages de programmation Le λ-calcul, à travers la « *correspondance de Curry-Howard* », a longtemps permis de fertiliser l'étude des langages de programmation avec les apports de la logique. Cependant, il est devenu courant de considérer que la théorie des langages de programmation pratiques doit souffrir maints écarts à cette dernière, en particulier lorsqu'intervient la notion d'ordre d'évaluation, avec les références par exemple.

On rejoint à ce sujet le travail de Zeilberger [Zei08a, Zei08b], qui montre comment retrouver certaines solutions *ad hoc* (en apparence) à partir des considérations théoriques que sont la polarisation et la focalisation : on pense aux *restrictions de valeur* que l'on impose afin de faire fonctionner le polymorphisme et les types intersection en appel par valeur. L'enjeu du programme de Zeilberger est de trouver un cadre théorique

plus adéquat pour les développements futurs en programmation typée.

Le système que l'on présente ici n'est pas sans intérêt au regard d'une telle entreprise. Tout d'abord, en comparaison avec le travail de Zeilberger, plus proche de la Ludique, les systèmes logiques avec lesquels on travaille sont conventionnels et la syntaxe est conforme à la tradition des langages de termes pour la logique.

Ensuite et en particulier, le traitement spécifique, à savoir l'introduction d'un décalage, que demandent les types intersection et universels en appel par valeur, ainsi que, par dualité, les types union et existentiels en appel par nom découle d'une même considération simple sur les comportements positifs (section 4), à savoir que leur propriété d'être engendrés par des valeurs n'est pas stable par intersection en général (section 5.2).

Motifs Dans le travail de Zeilberger [Zei08a] est mis en évidence le lien entre focalisation et *pattern-matching*. Ces caractéristiques y sont obtenues grâce à une configuration syntaxique, la *syntaxe abstraite focalisante de niveau supérieur*. On les retrouve ici en partie, sous des formes plus légères. Les motifs, par exemple, se retrouvent par simple convention d'écriture. On sera ammené à soutenir que cela peut-être préférable à des syntaxes donnant un statut formel aux motifs (section 2.4). La lisibilité induite par ces nos choix syntaxiques s'en ressent.

C'est donc en particulier une synthèse innovante des idées issues de ces travaux que permet le système L focalisant, avec tout le bénéfice de l'attrait et de l'accessibilité que permettent sa lisibilité et sa simplicité.

1.5 Pourquoi quatre connecteurs ?

D'aucuns pensent que du fait de l'équivalence en termes de prouvabilité, en logique classique, des deux conjonctions et des deux disjonctions, il est redondant de considérer, comme ici, les quatre connecteurs $(\otimes, \parr, \&, \oplus)$ à la fois. Il s'agit d'une idée reçue qui n'est pas justifiée dès lors que l'on s'intéresse au contenu procédural des preuves.

En effet, notre lecture distingue les quatre connecteurs selon leur caractère strict ou paresseux, et dès lors toute identification entre $\otimes$ et $\&$ ou entre $\oplus$ et $\parr$ serait abusive.

Le travail de Danos, Joinet et Schellinx [DJS95] considérait déjà la présence, dans la logique classique $\mathsf{LK}^{\mathrm{tq}}$, des quatre connecteurs issus de la logique linéaire, renommés pour l'occasion en $\wedge_{\mathrm{m}}, \vee_{\mathrm{m}}, \wedge_{\mathrm{a}}, \vee_{\mathrm{a}}$. 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 constitue donc un progrès.

1.6 La dualité du calcul reformulée

Puisque l'on fait de « paresseux » et de « strict » des propriétés des connecteurs, la dualité du calcul est reformulée ici de façon simple. Une différence importante de L avec la métaphore des automates finis — où dans $\langle \omega \parallel s \rangle$, ω et s n'habitent pas le même monde — est que les deux côtés de la commande sont de nature logique. La dualité du calcul provient de la possibilité que cela nous offre d'échanger les deux membres d'une commande $\langle t \parallel u \rangle$ (où t est le code et u l'environnement) afin d'obtenir une commande $\langle u \parallel t \rangle$ (où u est le code et t l'environnement) au comportement calculatoire similaire, mais dont le sens informatique est différent (section 2.3).

Cette formulation de la dualité du calcul est à comparer aux travaux précédents, dans lesquels la dualité est mise en évidence par un changement du protocole de réduction.

On s'efforcera de ne jamais s'écarter de la terminologie reçue sans de sérieuses raisons. Les notations utilisées ici sont principalement issues du calcul $\bar{\lambda}\mu\tilde{\mu}$ tel que présenté dans l'article de Curien et Herbelin [CH00], ou dans le mémoire d'habilitation de ce dernier [Her05].

2 Le système L

La présentation suivante de la syntaxe du système L présente quelques différences avec celle du $\bar{\lambda}\mu\tilde{\mu}$ [CH00] ou du système L d'Herbelin [Her08].

De façon similaire à ce dernier ou au calcul dual de Wadler, l'implication pourra se définir à partir de constructeurs plus basiques, dans une démarche analytique (l'implication en appel par valeur aura par exemple une définition différente

$$
\begin{array}{rcll}
\kappa & ::= & \alpha & \mid x \\
t & ::= & t_+ & \mid t_- \\
t_+ & ::= & x & \mid \mu\alpha.c \\
 & & \mid (t,t) & \mid \iota_i(t)\ (\text{pour } i \in \{1,2\}) \qquad (\otimes, \oplus_i) \\
 & & \mid () & \textit{(pas de constructeur pour } \mathbf{0}\textit{)} \qquad (\mathbf{1}, \mathbf{0}) \\
 & & \mid \mu\tilde{}\,(\kappa).c & \mid \{t\} \qquad (!, \downarrow) \\
t_- & ::= & \alpha & \mid \mu\tilde{x}.c \\
 & & \mid \mu(\kappa,\kappa).c\ ^{(*)} & \mid \mu\,(\iota_1\,(\kappa)\,.c \,|\, \iota_2\,(\kappa)\,.c) \qquad (\parr, \&) \\
 & & \mid \mu().c & \mid tp \qquad (\bot, \top) \\
 & & \mid \tilde{}\,(t) & \mid \mu\{\kappa\}.c \qquad (?, \uparrow) \\
c & ::= & \langle t_+ \parallel t_- \rangle & \mid \langle t_- \parallel t_+ \rangle
\end{array}
$$

$(*)$: avec $\mu(\kappa,\kappa').c$ non défini lorsque $\kappa = \kappa'$.

FIG. 1: La grammaire du système L.

de celle en appel par nom ; comme étudié par Laurent dans [Lau02]). Les additifs $\oplus$/& proviennent du calcul dual [Wad03]. Les multiplicatifs $\otimes/\parr$ proviennent de [Her05].

C'est dans les sections ultérieures qu'apparaît l'innovation principale, qui est d'associer à cette syntaxe une stratégie de réduction focalisante.

Definition 1. On considère deux ensembles dénombrables, respectivement $\{x, y, z \ldots\}$ de *variables positives* et $\{\alpha, \beta, \gamma \ldots\}$ de *variables négatives*.

On considère les ensembles de termes mutuellement définis par la grammaire fig. 1.

Dans $\langle t \parallel t' \rangle$, t est appelé le *contre-terme* de t' et t' le contre-terme de t.

Chacune des variables précédant un point dans cette syntaxe est liante. Sachant cela, on définit, en procédant comme à l'habitude, les relations d'α-équivalence sur les ensembles des termes t_+, t_- ou c d'une part, et la fonction $\mathcal{FV}(\cdot)$ des variables libres d'autre part.

L'ensemble des *termes positifs* noté $\mathcal{T}_+$, celui des *termes négatifs* noté $\mathcal{T}_-$ ainsi que celui des *commandes* noté $\mathcal{C}$ sont respectivement ceux des termes des classes t_+, t_- et c pris modulo α-équivalence.

Enfin, l'ensemble des *termes positifs clos* noté $\mathcal{T}_+^0$, celui des *termes négatifs clos* noté $\mathcal{T}_-^0$ ainsi que celui des *commandes closes* noté $\mathcal{C}^0$ sont respectivement ceux des éléments p de $\mathcal{T}_+$, $\mathcal{T}_-$ et $\mathcal{C}$ qui vérifient $\mathcal{FV}(p) = \emptyset$.

Contrairement au $\bar{\lambda}\mu\tilde{\mu}$ originel, l'accent est mis sur la distinction entre termes positifs et termes négatifs plutôt que sur la distinction entre code et environnement. Cela simplifie la syntaxe, puisque la redondance des connecteurs disparaît, sans que l'on perde en expressivité, car l'orientation des constructeurs entre alors en jeu.

Ainsi, au lieu du $\mu\alpha.c$ et du $\tilde{\mu}x.c$ des travaux sus-mentionnés, on peut lier une variable positive x avec μx et une variable négative α avec $\mu\alpha$. Le lien entre les deux notations est le suivant : si κ dénote une variable positive ou négative, alors $\langle \mu\kappa.c|$ se lira comme le $\mu\kappa.c$ du $\bar{\lambda}\mu\tilde{\mu}$, tandis que $|\mu\kappa.c\rangle$ se lira comme le $\tilde{\mu}\kappa.c$.

Ces différences syntaxiques sont mineures, la plus importante étant la partition naturelle des variables entre variables de code et variables d'environnement (co-variables) que l'on perd. Cette distinction peut se retrouver après typage dans un système bilatère ; ou alors au prix d'un doublement du nombre de constructeurs.

2.1 Systèmes de types

Definition 2 (Formules, formules positives et négatives)**.** On suppose donné un ensemble dénombrable d'*atomes* $\{X, Y \ldots\}$. On dénote par A, B les formules, par P, Q les formules polarisées positivement et par N, M les formules polarisées négativement. Elles sont engendrées par la grammaire fig. 2.

Definition 3 (Contextes, jugements)**.** $\Gamma, \Delta \ldots$ dénotent des *contextes,* des ensembles d'éléments de la forme $x : N$ ou $\alpha : P$. Les séquents du système L sont des jugements de la forme donnée fig. 2.

Les formules, formules positives et formules négatives du système L :

$$\begin{array}{rcl} A & ::= & P \mid N \\ P & ::= & X \mid A \otimes A \mid A \oplus A \mid \mathbf{1} \mid \mathbf{0} \mid !A \mid \exists X\, A \\ N & ::= & X^{\perp} \mid A \parr A \mid A \& A \mid \bot \mid \top \mid ?A \mid \forall X\, A \end{array}$$

Les jugements du système L :

$$\begin{array}{rcl} c : & (\vdash & \Gamma) \\ & \vdash & t_+ : P \mid \Gamma \\ & \vdash & t_- : N \mid \Gamma \end{array}$$

FIG. 2: Les formules et les jugements du système L.

MALL

Groupe identité Règles axiome, activation, coupure :

$$\frac{}{\vdash x : P \mid x : P^{\perp}}\ (ax+) \qquad \frac{}{\vdash \alpha : N \mid \alpha : N^{\perp}}\ (ax-)$$

$$\frac{c : (\vdash \kappa : A, \Gamma)}{\vdash \mu\kappa.c : A \mid \Gamma}\ (\mu) \qquad \frac{\vdash t : A \mid \Gamma \qquad \vdash u : A^{\perp} \mid \Delta}{\langle t \parallel u \rangle : (\vdash \Gamma, \Delta)}\ (\text{cut})$$

Groupe logique Règles tenseur, par, avec, plus :

$$\frac{\vdash t : A \mid \Gamma \qquad \vdash u : B \mid \Delta}{\vdash (t, u) : A \otimes B \mid \Gamma, \Delta}\ (\otimes) \qquad \frac{c : (\vdash \kappa : A, \kappa' : B, \Gamma)}{\vdash \mu(\kappa, \kappa').c : A \parr B \mid \Gamma}\ (\parr)$$

$$\frac{c : (\vdash \kappa : A, \Gamma) \qquad c' : (\vdash \kappa' : B, \Gamma)}{\vdash \mu\left(\iota_1(\kappa).c \mid \iota_2(\kappa').c'\right) : A \,\&\, B \mid \Gamma}\ (\&) \qquad \frac{\vdash t : A \mid \Gamma}{\vdash \iota_1(t) : A \oplus B \mid \Gamma}\ (\oplus_1) \qquad \frac{\vdash t : B \mid \Gamma}{\vdash \iota_2(t) : A \oplus B \mid \Gamma}\ (\oplus_2)$$

Extraction, généralisation :

$$\frac{\vdash t : A[P/X] \mid \Gamma}{\vdash \{t\} : \exists X\, A \mid \Gamma}\ (\exists) \qquad \frac{c : (\vdash \kappa : A, \Gamma)}{\vdash \mu\{\kappa\}.c : \forall X\, A \mid \Gamma}\ (\forall)\ (X \notin \mathcal{FV}(\Gamma))$$

Règles unité :

$$\frac{}{\vdash () : \mathbf{1} \mid}\ (\mathbf{1}) \qquad \frac{c : (\vdash \Gamma)}{\vdash \mu().c : \bot \mid \Gamma}\ \bot$$

$$\frac{}{\vdash tp : \top \mid \Gamma}\ (\top) \qquad \textit{pas de règle pour } \mathbf{0}$$

FIG. 3: Le fragment multiplicatif et additif MALL de la logique linéaire du second ordre.

$\mathsf{LK}_{\mathrm{pol}}$

MALL + le groupe structurel suivant :

Groupe structurel Règles décalages, affaiblissement, contraction :

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

$$\frac{\vdash t : A \mid \Gamma}{\vdash t : A \mid x : B, \Gamma}\ (w) \qquad \frac{\vdash t : A \mid x : B, y : B, \Gamma}{\vdash t[x/y] : A \mid x : B, \Gamma}\ (c)$$

FIG. 4: La logique classique constructive du second ordre $\mathsf{LK}_{\mathrm{pol}}$.

LL

MALL + le groupe structurel suivant :

Groupe structurel Règles promotion, déréliction, affaiblissement, contraction :

$$\frac{\vdash t : A \mid \Gamma}{\vdash \text{`}(t) : ?A \mid \Gamma}\ (?d) \qquad \frac{c : (\vdash \kappa : A, ?\Gamma)}{\vdash \mu\text{`}(\kappa).c : !A \mid ?\Gamma}\ (!)$$

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

$$\frac{\vdash t : A \mid \Gamma}{\vdash t : A \mid x : ?B, \Gamma}\ (?w) \qquad \frac{\vdash t : A \mid x : ?B, y : ?B, \Gamma}{\vdash t[x/y] : A \mid x : ?B, \Gamma}\ (?c)$$

FIG. 5: La logique linéaire LL du second ordre.

Dans $\vdash t_+ : P \mid \Gamma$ (resp. dans $\vdash t_- : N \mid \Gamma$), 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 *bénitier* ou de *formule focalisée*.

Definition 4 (Dualité)**.** À chaque formule positive P (respectivement chaque formule négative N) correspond *formule duale* $P^\perp$ négative (resp. $N^\perp$ positive) donnée par :

$$\begin{array}{cc}
(X)^\perp \equiv X^\perp & (X^\perp)^\perp \equiv X \\
(A \otimes B)^\perp \equiv A^\perp \parr B^\perp & (A \parr B)^\perp \equiv A^\perp \otimes B^\perp \\
(A \oplus B)^\perp \equiv A^\perp \,\&\, B^\perp & (A \,\&\, B)^\perp \equiv A^\perp \oplus B^\perp \\
\mathbf{1}^\perp \equiv \bot & \bot^\perp \equiv \mathbf{1} \\
\mathbf{0}^\perp \equiv \top & \top^\perp \equiv \mathbf{0} \\
(\exists X\, A)^\perp \equiv \forall X\, A^\perp & (\forall X\, A)^\perp \equiv \exists X\, A^\perp \\
(!A)^\perp \equiv ?A^\perp & (?A)^\perp \equiv !(A^\perp)
\end{array}$$

Definition 5. 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 récur-

rence sur la structure de A. Les cas importants sont $X\,[P/X] = P$ et $X^\perp\,[P/X] = P^\perp$.

Definition 6. On donne les règles de typage du système L dans MALL (fig. 3), $\mathsf{LK}_{\mathrm{pol}}$ (fig. 4) et LL (fig. 5) monolatères et du second ordre.

Dans le groupe identité, 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 $\vdash v : A \mid \Gamma$.

Proposition 7. On dérive du groupe identité les règles suivantes :

$$\frac{\vdash t : N \mid \Gamma}{\langle x \parallel t\rangle : (\vdash x : N, \Gamma)}\ (\text{désactivation } -)$$

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

2.2 Discussions sur la syntaxe

2.2.1 Avec

On a choisi une syntaxe de *pattern-matching* pour le constructeur & pour des raisons expliquées en 2.4. Il est cependant vrai que $(t|t')$ est plus court pour certains usages, comme pour dénoter la paire paresseuse. En réalité, peu importe car on pourra alors définir le sucre suivant :

$$(t|t') = \mu\left(\iota_1(\kappa).\langle t \parallel \kappa\rangle \,\middle|\, \iota_2(\kappa).\langle t' \parallel \kappa\rangle\right)$$

2.2.2 Décalages

Un terme négatif peut s'employer en place d'un terme positif : un tel usage est un décalage, c'est-à-dire un changement de polarité, des négatifs vers les positifs ($\downarrow$). Réciproquement, un terme positif peut être transformé en un terme négatif, mais cela n'est pas gratuit : un tel décalage des positifs vers les négatifs ($\uparrow$) a pour effet calculatoire de retarder l'exécution du terme.

Conformément à LC, et contrairement aux systèmes polarisés comme la Ludique ou le Calculus of Unity de Zeilberger, ces décalages sont implicites et n'apparaissent pas dans les types.

Ce n'est pas contradictoire avec le fait que la syntaxe admette un connecteur $\{\cdot\}$ correspondant au décalage $\downarrow$. Cette présence est peu discutable, car elle est à la fois gratuite et nécessaire. *Nécessaire :* on verra en section 5.2 que le décalage est indispensable pour le traitement de la quantification. (C'est de là que provient la notation : $\{\cdot\}$ marque l'extraction dans le système $\mathbb{F}$). *Gratuite :* le décalage $\downarrow$ n'est jamais qu'un tenseur unaire ; en l'absence de $\{\cdot\}$, on peut remplacer celui-ci par $(\cdot, ())$.

2.2.3 Exponentielles

Notre choix d'une syntaxe de « pattern-matching » pour le bien-sûr ! en fait le seul connecteur positif à être écrit ainsi — tous les autres pattern-matchings sont négatifs. Il nous faut donc expliquer ce choix.

Il découle du fait que la règle « η » :

$$t \to_\eta \mu'(\kappa).\langle t \parallel '(\kappa)\rangle$$

a un sens en logique linéaire au moins lorsque t est une valeur (notion définie plus loin), c'est-à-dire que la réversibilité du ! est prouvable sous cette condition. *A contrario,* le choix inverse, qui viserait à attribuer l'écriture « pattern-matching » à son dual pourquoi-pas ? n'est pas acceptable, car il n'y a pas de notion de réversibilité du ? dans LL.

2.2.4 Séquents bilatéraux, séquents droits

Rappelons que la littérature admet deux traditions vis-à-vis des séquents : la tradition historique est due à Gentzen propose des séquents bilatéraux, c'est-à-dire avec des formules des deux côtés du tourniquet $\vdash$. Plus récemment, Girard a proposé une écriture avec des séquents droits, c'est-à-dire où toutes les formules sont à la droite du tourniquet. Cette notation est justifiée car les règles des calculs des séquents à la Gentzen sont souvent symétriques : les séquents bilatéraux $\Gamma \vdash \Delta$ sont repliés en des séquents droits $\vdash \Gamma^\perp, \Delta$ et la négation n'est plus un connecteur mais devient un morphisme involutif entre formules positives et négatives. Cela a pour avantage de diminuer le nombre de règles.

Le reste de l'article usera de la seconde tradition par souci de simplification. La syntaxe permet cependant les deux écritures, et il peut être utile de clarifier le lien qu'elles entretiennent.

La tradition de Gentzen implique de faire la distinction entre la commande $\langle t \parallel u\rangle$ et la commande $\langle u \parallel t\rangle$. Cela permet la distinction entre le *code* (ce qui est à la droite du tourniquet) et

l'*environnement* (ce qui est à la gauche du tourniquet). En héritage du $\bar{\lambda}\mu\tilde{\mu}$ ou des machines abstraites comme la machine de Krivine [Kri04], dans $\langle t \parallel u \rangle$, t représente le code et u l'environnement.

La tradition de Girard correspond à quotienter la syntaxe par l'équivalence $\langle t \parallel u \rangle \equiv \langle u \parallel t \rangle$. L'interprétation informatique termes de machines abstraites est perdue, précisément car cela revient à raisonner modulo la dualité du calcul. Puisque cette tradition est retenue pour la suite, on admet dans la syntaxe cette équivalence.

$\mathsf{LK}_{\mathrm{pol}}$ et LL bilatères Écrire les règles de $\mathsf{LK}_{\mathrm{pol}}$ et LL suivant la tradition de Gentzen entraîne un doublement du nombre de règles d'inférence. Cet exercice est donc laissé au lecteur. Quelques indications nécessaire suivent.

Le contexte à la gauche du tourniquet devient un ensemble d'éléments de la forme $x : P$ ou $\alpha : N$. En outre, le morphisme de négation $\cdot^{\perp}$ est remplacé par un connecteur en bonne et due forme noté $\sim$, et tel que la polarité de $\sim A$ soit l'opposé de celle de A. Il est donc nécessaire d'introduire un nouveau constructeur qui marque l'échange entre code et environnement : $[\cdot]$. Il est muni des réductions suivantes :

$$\begin{aligned}
\langle [V] \parallel \mu\,[\kappa]\,.c \rangle &\to_\beta c\,[V/\kappa] \\
\langle \mu\,[\kappa]\,.c \parallel [V] \rangle &\to_\beta c\,[V/\kappa] \\
\langle [t] \parallel u \rangle &\to_\sigma \left\langle \mu x.\langle [x] \parallel u \rangle \,\middle\|\, t \right\rangle \\
\langle u \parallel [t] \rangle &\to_\sigma \left\langle t \,\middle\|\, \mu x.\langle u \parallel [x] \rangle \right\rangle
\end{aligned}$$

Cette négation vient avec les règles suivantes :

$$\frac{\Gamma \vdash t_+ : P \mid \Delta}{\Gamma \mid [t_+] : \sim P \vdash \Delta} \qquad \frac{\Gamma \mid t_+ : N \vdash \Delta}{\Gamma \vdash [t_+] : \sim N \mid \Delta}$$

$$\frac{c : (\Gamma \vdash x : N, \Delta)}{\Gamma \mid \mu\,[x].c : \sim N \vdash \Delta} \qquad \frac{c : (\Gamma, x : P \vdash \Delta)}{\Gamma \vdash \mu\,[x].c : \sim P \mid \Delta}$$

Il s'agit de la négation linéaire ; en particulier, $\sim\sim P$ est isomorphe à P puisque $[[\cdot]]$ n'a d'autre comportement calculatoire que celui d'un tenseur unaire, et $\sim$ vérifie les isomorphismes attendus, comme $\sim(A \otimes B) \simeq \sim A \parr \sim B$.

2.2.5 tp/$\top$

Les unités admettent une interprétation informatique claire. À côté de l'unité du tenseur, $\mathbf{1}$, habité par un unique élément noté de façon évidente $()$, il y a le neutre du avec : $\top$. On donne ce type à la *constante toplevel* $|\mathsf{tp}\rangle$.

Ce terme négatif a le comportement calculatoire suivant : une fois son contre-terme positif évalué, le calcul ne se poursuit pas. En Ludique, le toplevel correspond donc au dessein vide, le sconse. Ceci justifie la règle de typage de tp.

Une conséquence directe est que l'on retrouve la définition de l'opérateur *abort* par Ariola, Herbelin et Sabry [AHS04] :

On pose $\mathcal{A}(t_+) \stackrel{\text{def}}{=} \mu\_.\langle t_+ \parallel \mathsf{tp} \rangle$. De la règle $\top$ est alors conséquence la règle suivante :

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

Dualement, $\langle \mathsf{tp}|$ désigne un terme qui ne définit aucune valeur. En appel par nom, cela ne correspond à un calcul erroné que lorsque l'on tente d'y accéder. On peut donc penser pour l'interprétation calculatoire de $\langle \mathsf{tp}|$ au pointeur nul.

2.2.6 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 permise vis-à-vis du calcul original concerne la polarité des atomes, ce qui est une restriction qui n'existe pas dans le calcul des séquents LL d'origine. Par exemple, dans ce dernier, prouver $\forall X(X \multimap X)$ revient à prouver à la fois $\forall X(X \multimap X)$ et $\forall X(X^{\perp} \multimap X^{\perp})$ dans notre système.

Toutes les formules non-atomique possédant une polarité, cette contrainte sur les atomes est légitime. Elle est également superficielle.

En effet, étendons notre système avec des atomes « neutres », que l'on note $X^?$, et que l'on peut indifféremment spécialiser en un atome positif ou en un atome négatif. Obtenir une dérivation contenant un tel atome neutre équivaut alors à obtenir les dérivations correspondant à chaque choix de polarité pour cet atome. Une telle dérivation est dont la superposition de deux preuves, et les termes issus de celles-ci ne différeront que si ces preuves font apparaître la règle axiome pour $X^?$. Cette règle devient alors la superposition des règles axiome positive et négative, que l'on écrira : $\overline{\vdash \kappa : X^? \mid \kappa : X^{?\perp}}$. Cela signifie qu'avec la convention pour laquelle $\kappa, \kappa', \ldots$ dénotent des variables qui peuvent au choix être

positives ou négatives, un terme unique représente cette superposition de preuves.

Par exemple, une preuve de $\forall X(X^? \multimap X^?)$ sera obtenue lorsque l'on pourra à la fois prouver $\forall X(X \multimap X)$ et $\forall X(X^\perp \multimap X^\perp)$. Ces deux preuves sont respectivement $\mu\{(x,\alpha)\}.\langle x \parallel \alpha\rangle$ et $\mu\{(\alpha,x)\}.\langle \alpha \parallel x\rangle$. La preuve de $\forall X(X^? \multimap X^?)$ peut donc s'écrire $\mu\{(\kappa,\kappa')\}.\langle \kappa \parallel \kappa'\rangle$.

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* $\otimes$, $\parr$, & et $\oplus$ sont respectivement $(\cdot,\cdot)$, $\mu(\cdot,\cdot).\cdot$, $\mu\,(\iota_1\,(\cdot)\,.\cdot\,|\,\iota_2\,(\cdot)\,.\cdot)$ et $\iota_i(\cdot)$. Les *connecteurs* $\otimes$, $\parr$, & 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 $\otimes$: $\langle(\cdot,\cdot)|$ est le constructeur de la paire stricte et $|(\cdot,\cdot)\rangle$ est le *destructeur* de la disjonction paresseuse.
- Constructeur $\parr$: $\langle\mu(\cdot,\cdot).\cdot|$ est le constructeur de la disjonction paresseuse[4] et $|\mu(\cdot,\cdot).\cdot\rangle$ est le destructeur de la paire stricte.
- Constructeur & : $\langle\mu\,(\iota_1\,(\cdot)\,.\cdot\,|\,\iota_2\,(\cdot)\,.\cdot)|$ (ou encore : $\langle(\cdot|\cdot)|\rangle$) est le constructeur de la paire paresseuse et $|\mu\,(\iota_1\,(\cdot)\,.\cdot\,|\,\iota_2\,(\cdot)\,.\cdot)\rangle$ est le destructeur de la disjonction stricte.
- Constructeur $\oplus$: $\langle\iota_i(\cdot)|$ 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 $\parr$ (resp. &) pour l'involution $\cdot^\perp$. Le connecteur $\otimes$ (resp. $\oplus$) est le dual du connecteur $\parr$ (resp. &) pour la dualité du calcul.

2.4 Motifs

Les constructeurs réversibles sont donnés sous la forme de *pattern-matching*. Le lien entre réversibilité et *pattern-matching* a été mis en évidence indépendemment par Zeilberger [Zei08a]. On soutient ici qu'il est commode de travailler dans une syntaxe avec *pattern-matching* par une convention d'écriture plutôt que par un formalisme.

*Notation :*On se permet de composer les motifs du μ. Cela permet une concision attrayante. Sans rentrer dans une définition exhaustive (c'est là l'intérêt), on pourra utiliser par exemple la notation :

$$\mu\,(\iota_1\,(x,y)\,.c_1\,|\,\iota_2\,(z)\,.c_2)$$

pour remplacer :

$$\mu\,\big(\iota_1\,(a)\,.\langle a \parallel \mu(x,y).c_1\rangle \,\big|\, \iota_2\,(z)\,.c_2\big)$$

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

$$\mu\text{`}\,(x,y).\langle y \parallel \text{`}\,(t)\rangle$$

au lieu de :

$$\mu\text{`}\,(z).\Big\langle z \Big\| \mu(x,y).\langle y \parallel \text{`}\,(t)\rangle\Big\rangle$$

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

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

$$\frac{\vdash v : A \mid x : N, y : M, \Gamma}{\vdash v : A \mid (x,y) : N \parr M, \Gamma}\ (\parr)$$

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 en compte la réversibilité du $\parr$.

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

$$\mu x.\langle x \parallel t\rangle =_\eta t$$

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

$$\mu x.\langle (x,x) \parallel \alpha\rangle$$

[4] Comme remarqué par Filinski [Fil89] et Zeilberger [Zei08a], il ne s'agit pas de la somme paresseuse au sens de Haskell.

et :

$$\mu(y,z).\langle((y,z),(y,z)) \parallel \alpha\rangle$$

modulo α-conversion, ce qui exprime également la réversibilité du $\mathfrak{P}$.

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]). Zeilberger fait des patterns formels une caractéristique centrale de son calcul polarisé [Zei08a], ce qui est un héritage de la Ludique. Par ailleurs, dans [Wad04], Wadler propose un calcul de purs motifs pour la logique linéaire.

Malheureusement, ce dernier travail laisse aussi préfigurer la bureaucratie que peut entraîner une telle *simplification* du système L.

Le travail de Zeilberger évite ces difficultés dans le cadre d'une syntaxe d'ordre supérieure, mais il ne s'agit malheureusement pas d'une syntaxe de termes au sens finitaire du terme.

Pour la plupart des usages, un tel calcul avec des motifs de première classe est suffisant. On peut très bien obtenir les avantages des patterns de façon informelle.

En effet, d'une part, on peut faire de la construction de motifs une convention de notation comme dans la notation 2.4. D'autre part, on peut permettre aux contextes de contenir des motifs : il suffit de considérer cela comme une astuce informelle de dérivation.

Ce que l'on propose est un *in*formalisme : cela se fait donc à bureaucratie constante. Le quotient supplementaire permis par un usage formel des patterns sera en outre mis en évidence dans [CHM09].

3 Protocole de réduction focalisant

La section s'attache à définir sur L un protocole un réduction *focalisant* qui est essentiellement une adaptation à un cadre non typé de celui de LC donné par Girard. On nommera *système* L *focalisant* le système L muni de cette stratégie de réduction. Rappelons qu'une particularité de LC est de pouvoir distinguer une formule positive à travers un dispositif syntaxique (le bénitier) ; cette formule est alors dite focalisée.

La discipline imposée par le bénitier a pour homologue en informatique une tradition ancienne due à Plotkin [Plo75] : faire intervenir dans la définition du protocole un sous-ensemble de termes, les *valeurs*. Cette remarque due à Curien et Herbelin [CH00] semble être passée inaperçue. La méthode de Plotkin, à première vue non logique, correspond donc à la focalisation. Zeilberger le redécouvre lorsqu'il associe les motifs et la focalisation en Ludique [Zei08a].

Malgré l'âge de LC et le caractère très informatique du protocole reformulé, c'est la première fois que la présentation qui suit fait surface. (On présente en fait $\mathsf{LK}_{\mathrm{pol}}$, que l'on définit comme la variante de LC qui possède les quatre connecteurs issus de la logique linéaire. On retrouve LC en codant $\vee$ et $\wedge$ à la manière de l'article original.)

Comparaison à d'autres syntaxes La question d'une syntaxe pour LC, posée par Girard dans [Gir91], est ancienne. Elle a trouvé une première réponse avec les réseaux polarisés de Laurent [Lau02]. Mais l'éclairage que L apporte sur LC est tout à fait différent de celui apporté par le travail de Laurent. En premier lieu car dans ce dernier, la lecture se fait à travers une traduction vers LLP. L'ordre d'évaluation est alors codé au moyen des boites des réseaux polarisés. Le lien avec la technique de Plotkin n'apparaît pas.

Par ailleurs, le calcul $\lambda\mu$ est déjà une syntaxe pour la version négative de LC : LKT, mais ce n'est pas satisfaisant : $\lambda\mu$ n'a pas « *la saine naïveté du* λ*-calcul* » [Gir07]. En outre, il ne permet pas d'accéder à la négation linéaire.

L focalisant est donc la première syntaxe à décrire précisément LC.

Paires critiques Le protocole de LC résout les paires critiques en donnant la priorité au côté positif des coupures. Cependant, ainsi que le remarque Girard [Gir91], cela ne résout pas toutes les paires critiques : il reste celle associée à la conjonction positive. C'est une banalité que de dire que l'ordre d'évaluation de la paire stricte importe.

LC résout cela en restreignant la paire (t,u) aux t,u *centraux*, c'est-à-dire qui se comportent comme des valeurs. Notre choix est d'imposer un ordre d'évaluation par défaut, de gauche à droite, du constructeur $\otimes$ (règle $\rightarrow_\sigma$). Cette extension est triviale : une formulation équivalente serait de considérer que (t,u) est défini dans le

cas général :

$$(t,u) \stackrel{\text{def}}{=} \mu\alpha.\Big\langle t \,\Big\|\, \mu x.\big\langle u \,\big\|\, \mu y.\langle (x,y) \,\|\, \alpha\rangle\big\rangle\Big\rangle$$

Le but de la syntaxe est d'être lisible et écrivable ; les règles $\to_\sigma$ visent ce but.

La notation $\to_\sigma$ est héritée du Calcul Dual de Wadler [Wad03], qui emploie des règles similaires dans un cadre voisin du $\bar{\lambda}\mu\tilde{\mu}$.

3.1 Valeurs et commandes sans coupures

Definition 8 (Valeurs)**.** On définit les classes des *valeurs* et des *valeurs positives*, cas particulier des termes et des termes positifs, par la grammaire fig 6. (On décide donc par convention que tout terme négatif est une valeur.)

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

On définit ensuite la classe des *négatifs bloquants* comme étant l'ensemble des termes négatifs qui ne sont d'aucune des formes suivantes : $\mu x.c$, $`(t)$ avec $t \notin \mathbb{V}$. W dénote un négatif bloquant.

Definition 9. Une commande est une *coupure* si elle n'est d'aucune des formes suivantes :

$$\langle V_+ \,\|\, \alpha\rangle$$
$$\langle V_+ \,\|\, tp\rangle$$
$$\langle x \,\|\, W\rangle$$

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.

3.2 Le protocole

Definition 10 (Réduction de tête)**.** On définit fig. 7 les relations suivantes sur $\mathcal{C}$: les μ-réductions $\to_{\mu_-}$ et $\to_{\mu_+}$, la β-réduction $\to_\beta$, la σ-réduction $\to_\sigma$ ainsi que la réduction de tête $\to_h$. Elles sont données avec $\langle t \,\|\, u\rangle \equiv \langle u \,\|\, t\rangle$: dans une configuration bilatère, on considérera également les règles symétriques.

Remark 11. On peut reformuler $\to_{\mu_-} \cup \to_{\mu_+}$ par :

$$\langle \mu\kappa.c \,\|\, V\rangle \to_\mu c[V/\kappa]$$

Definition 12 (Réduction contextuelle)**.** On pose $c \to c'$ si on peut obtenir c' à partir de c en appliquant la réduction de tête $\to_h$ sur une sous-commande.

Lemma 13. Si t_+ n'est pas une valeur positive ou t_- n'est pas un négatif bloquant, alors $\langle t_+ \,\|\, t_-\rangle$ est un redex pour $\to_h$.

Démonstration. Si t_+ n'est pas une valeur positive, alors $\langle t_+ \,\|\, t_-\rangle$ est un redex pour $\to_{\mu_+}$ ou pour $\to_\sigma$. Si t_+ est une valeur positive mais t_- n'est pas un négatif bloquant : si celui-ci est de la forme $`(t'_+)$, alors nous avons un redex pour $\to_\sigma$. Sinon, c'est qu'il est de la forme $\mu x.c$ et alors nous avons un redex pour $\to_\mu$. □

Definition 14. La relation $\to_\eta$ est définie sur $\mathcal{C}$ en fig. 8.

3.3 Subject reduction

On énonce ici une propriété de *subject reduction* pour LL et $\mathsf{LK}_{\mathrm{pol}}$.

Proposition 15 (Subject reduction)**.** Soit $c \to_h c'$. Si c est typable dans LL (respectivement $\mathsf{LK}_{\mathrm{pol}}$) dans un contexte Γ, alors c' est typable dans LL (resp. $\mathsf{LK}_{\mathrm{pol}}$) dans le contexte Γ.

Démonstration. Le résultat est évident bien que potentiellement bureaucratique (ce que l'on évitera ici). On considère le cas de LL qui est le moins libéral des deux. Par cas sur l'origine de $c \to_h c'$. (1) Il s'agit de $\langle \mu\kappa.c \,\|\, V\rangle \to_\mu c[V/\kappa]$. Si κ ne fait pas l'objet de contractions ou d'affaiblissements dans c, alors on peut effectuer la substitution linéaire des règles axiome pour κ dans la dérivation de c par la dérivation de V. Sinon, cela signifie que κ est de type $?A$ dans Γ, et donc V est de type $!A^\perp$. On en déduit que V, qui par hypothèse est une valeur, est soit issu d'une règle axiome, soit issu d'une promotion. Dans les deux cas, le contexte de V est de la forme $?\Delta$. Cela permet de procéder à la substitution des règles axiome pour κ dans la dérivation de c, des règles structurelles sur $?\Delta$ remplaçant celles sur $?A$. (2) Il s'agit de $c \to_\beta c'$. On effectue une transformation locale de la preuve qui nous permet de nous ramener au cas précédent. Par exemple, pour le cas $c = \langle (V,V') \,\|\, \mu(\kappa,\kappa').c\rangle$, alors $\big\langle V \,\big\|\, \mu\kappa.\langle V' \,\|\, \mu\kappa'.c\rangle\big\rangle \to_h^2 c'$. Le membre de gauche est typable dans Γ, et on est ainsi ramené à appliquer deux fois le cas précédent. (3) Il s'agit de $c \to_\sigma c'$. Facile. □

Proposition 16. Soit $c \to_\eta c'$. Si c est typable dans un contexte Γ, alors c' l'est aussi.

Valeurs, valeurs positives :

$$\begin{array}{rcl} V & = & V_+ \mid t_- \\ V_+ & = & x \mid (V,V) \mid \iota_i(V) \mid \mu\tilde{}(\kappa).c \mid \{V\} \mid () \end{array}$$

FIG. 6: Les valeurs $\mathbb{V}$ du système L focalisant.

μ-réduction

$$\begin{array}{rcl} \langle \mu\alpha.c \parallel t_- \rangle & \to_{\mu_-} & c[t_-/\alpha] \\ \langle \mu x.c \parallel V_+ \rangle & \to_{\mu_+} & c[V_+/x] \end{array}$$

β-réduction Dans le cas où les polarités des Vs et des κs ne se correspondent pas, la relation $\to_\beta$ n'est pas définie.

$$\begin{array}{rcll} \langle (V,V') \parallel \mu(\kappa,\kappa').c \rangle & \to_\beta & c\,[V,V'/\kappa,\kappa'] & (\otimes/\parr) \\ \langle \iota_1(V) \parallel \mu\,(\iota_1\,(\kappa)\,.c \mid \iota_2\,(\kappa')\,.c') \rangle & \to_\beta & c\,[V/\kappa] & (\oplus_1/\&) \\ \langle \iota_2(V) \parallel \mu\,(\iota_1\,(\kappa)\,.c \mid \iota_2\,(\kappa')\,.c') \rangle & \to_\beta & c\,[V/\kappa'] & (\oplus_2/\&) \\ \langle () \parallel \mu().c \rangle & \to_\beta & c & (\mathbf{1}/\bot) \\ \langle \mu\tilde{}(\kappa).c \parallel \tilde{}(V) \rangle & \to_\beta & c\,[V/\kappa] & (!/?) \\ \langle \{V\} \parallel \mu\{\kappa\}.c \rangle & \to_\beta & c\,[V/\kappa] & (\downarrow/\uparrow) \end{array}$$

σ-réduction Si la partie gauche des commandes des membres de gauche qui suivent ne sont pas des valeurs (dans le $4^{\text{ème}}$ cas : pas de la forme $\tilde{}(V)$), alors on pose, pour des variables κ, κ', x non libres dans u_- et V_+ :

$$\begin{array}{rcl} \langle (t,t') \parallel u_- \rangle & \to_\sigma & \left\langle t \,\middle\|\, \mu\kappa.\left\langle t' \,\middle\|\, \mu\kappa'.\langle (\kappa,\kappa') \parallel u_- \rangle \right\rangle \right\rangle \\ \langle \iota_i(t) \parallel u_- \rangle & \to_\sigma & \langle t \parallel \mu x.\langle \iota_i(x) \parallel u_- \rangle \rangle \\ \langle \{t\} \parallel u_- \rangle & \to_\sigma & \langle t \parallel \mu x.\langle \{x\} \parallel u_- \rangle \rangle \\ \langle \tilde{}(t) \parallel V_+ \rangle & \to_\sigma & \left\langle t \,\middle\|\, \mu x.\left\langle \tilde{}(x) \,\middle\|\, V_+ \right\rangle \right\rangle \end{array}$$

Réduction de tête

$$\to_h \;=\; \to_{\mu_-} \cup \to_{\mu_+} \cup \to_\beta \cup \to_\sigma$$

FIG. 7: Protocole de réduction focalisant pour le système L avec la convention monolatère $\langle t \parallel u \rangle \equiv \langle u \parallel t \rangle$.

L'expansion η sur les commandes. Pour des variables κ, κ' faisant concorder les polarités.

$$\begin{array}{rcll}
\langle (t, t') \parallel t_- \rangle & \to_\eta & \left\langle (t, t') \,\middle\|\, \mu(\kappa, \kappa').\langle (\kappa, \kappa') \parallel t_- \rangle \right\rangle & (\eta_{\parr}) \\
\langle \iota_i(t) \parallel t_- \rangle & \to_\eta & \left\langle \iota_i(t) \,\middle\|\, \mu\left(\iota_1(\kappa).\langle \iota_1(\kappa) \parallel t_- \rangle \,\middle|\, \iota_2(\kappa').\langle \iota_2(\kappa') \parallel t_- \rangle\right) \right\rangle & (\eta_{\&}) \\
\langle \{t\} \parallel t_- \rangle & \to_\eta & \left\langle \{t\} \,\middle\|\, \mu\{\kappa\}.\langle \{\kappa\} \parallel t_- \rangle \right\rangle & (\eta_{\uparrow}) \\
\langle () \parallel t_- \rangle & \to_\eta & \left\langle () \,\middle\|\, \mu().\langle () \parallel t_- \rangle \right\rangle & (\eta_{\bot}) \\
\langle V_+ \parallel \text{`}(t) \rangle & \to_\eta & \left\langle \mu\text{`}(\kappa).\langle V_+ \parallel \text{`}(\kappa) \rangle \,\middle\|\, \text{`}(t) \right\rangle & (\eta_{!})
\end{array}$$

FIG. 8: L'expansion η.

Démonstration. Cela provient de la réversibilité des connecteurs $\parr, \&, \forall, \bot$ et de la semi-réversibilité de !. □

3.4 Termes mal formés

Proposition 17 (Termes mal-formés, caractérisation)**.** On désigne par $t_\otimes$, $t_\oplus$, $t_\downarrow$, $t_{\mathbf{1}}$, $t_!$, $t_{\parr}$, $t_\&$, $t_\uparrow$, $t_\bot$, et $t_?$ des termes respectivement de la forme (t, t'), $\iota_i(t)$, $\{t\}$, $()$, $\mu\text{`}(\kappa).c$, $\mu(\kappa, \kappa').c$, $\mu(\iota_1(\kappa).c \,|\, \iota_2(\kappa').c')$, $\mu\{\kappa\}.c$, $\mu().c$ et $\text{`}(t)$. Soit c une commande de la forme $\langle t_\circledast \parallel t_{\circledast'} \rangle$ avec $\circledast$ parmi $\otimes, \oplus, \downarrow, \mathbf{1}, !$ et $\circledast'$ parmi $\parr, \&, \uparrow, \bot, ?$. Les deux propriétés suivantes sont équivalentes :

1. c n'est pas un redex pour $\to_h$.
2. $t_\circledast$ est une valeur positive et $t_{\circledast'}$ est un négatif bloquant, avec l'une des suivantes :
   a) $\circledast' \neq \circledast^\bot$;
   b) Les polarités des sous-termes du $t_\otimes$, du $t_\oplus$, du $t_\downarrow$ ou du $t_?$ ne correspondent pas à celles des variables liées par le $t_{\parr}$, le $t_\&$, le $t_\uparrow$ ou le $t_!$.

Démonstration. ($1 \Rightarrow 2$) Si $t_\circledast$ n'est pas une valeur positive ou $t_{\circledast'}$ n'est pas un négatif bloquant, alors c est un redex pour $\to_h$. Sinon, si $\circledast$ et $\circledast'$ sont duaux et que les polarités des variables concordent, alors c est un redex pour $\to_\beta$. ($2 \Rightarrow 1$) c étant de la forme $\langle t_\circledast \parallel t_{\circledast'} \rangle$, elle n'est pas un redex pour $\to_\mu$. Comme $\circledast$ et $\circledast'$ ne sont pas duaux et que les polarités des variables ne concordent pas, c n'est pas un redex pour $\to_\beta$. Enfin, comme $t_\circledast$ est une valeur positive et $t_{\circledast'}$ est un négatif bloquant, c n'est pas un redex pour $\to_\sigma$. □

Definition 18 (Termes mal-formés)**.** Une commande de la forme $\langle t_\circledast \parallel t_{\circledast'} \rangle$ qui satisfait l'une des deux propriétés équivalentes de la proposition 17 est dite *mal-formée*.

Remark 19. La commande suivante n'est donc pas mal-formée :

$$\left\langle (\mu\_.c, \mathtt{toto}) \,\middle\|\, \tilde{\mu}(\alpha, x).c' \right\rangle$$

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 qui est demandé pour la compatibilité avec les règles η.

Proposition 20. Pour toute commande c, c est un redex pour $\to_h$ si et seulement si c est une coupure et c n'est pas mal-formée.

Démonstration. ($\Rightarrow$) Par définition, une commande qui n'est pas une coupure n'est pas un redex pour $\to_h$, et une commande qui est un redex pour $\to_h$ n'est pas mal-formée. ($\Leftarrow$) Soit $\langle t_+ \parallel t_- \rangle$ une commande qui n'est pas mal formée et qui n'est pas un redex pour $\to_h$. Par définition, elle n'est pas de la forme $\langle t_\circledast \parallel t_{\circledast'} \rangle$. En outre, t_+ est une valeur positive et t_- est un négatif bloquant. Cela signifie que t_+ est de la forme x ou que t_- est de la forme α ou $\mathtt{tp}$. Donc $\langle t_+ \parallel t_- \rangle$ n'est pas une coupure. □

3.5 Confluence

Proposition 21. $\to_h$ n'admet pas de paire critique, *i.e.* si $c \to_h c'$ et $c \to_h c''$, alors $c' = c''$.

Démonstration. Par construction de $\to_h$. □

Proposition 22. La relation $\to^*$ est confluente, *i.e.* si $c \to^* c'$ et $c \to^* c''$, alors il existe c''' tel que $c' \to^* c'''$ et $c'' \to^* c'''$.

Démonstration. Par une adaptation aisée de la méthode de Tait, en conséquence de la proposition 21. □

Enfin, l'élimination de tête admet la compatibilité suivante avec $\to_\eta$:

Proposition 23. Soient c, c' et c'' trois commandes. Si $c \to_\eta c'$ et $c \to_h c''$, alors $c' \to^2 c''$.

Démonstration. Si $c \to_h c''$ provient de $c \to_{\mu_+} c''$ ou de $c \to_\beta c''$, alors $c' \to^2 c''$ provient de $c' \to_\beta c$. Sinon, alors il s'agit de $c \to_\sigma c''$, auquel cas $c' \to^2 c''$ provient d'une réduction $\to_\sigma$ et d'une réduction $\to_\beta$ sur une sous-commande. Le cas $c \to_{\mu_-} c''$ est exclu par définition de $\to_\eta$. □

3.6 Exemples

On s'intéresse à l'implication. On se place donc dans le cas bilatère, en notant avec Curien-Herbelin [CH00] v le code et e l'environnement. On pose :

$$\begin{aligned}
A \to B &\stackrel{\text{def}}{=} {\sim}A \parr B \\
\langle \lambda\kappa.v| &\stackrel{\text{def}}{=} \left\langle \mu([\kappa], \kappa').\langle v \,\|\, \kappa' \rangle \right| \quad (\kappa' \notin \mathcal{FV}(t)) \\
|v \cdot e\rangle &\stackrel{\text{def}}{=} |([v], e)\rangle \\
\langle v\, v'| &\stackrel{\text{def}}{=} \left\langle \mu\kappa.\langle v \,\|\, v' \cdot \kappa \rangle \right|
\end{aligned}$$

On dérive alors :

$$\frac{\Gamma, \kappa : A \vdash v : B \mid \Delta}{\Gamma \vdash \lambda\kappa.v : A \to B \mid \Delta}\ (\vdash \to)$$

$$\frac{\Gamma \vdash v : A \mid \Delta \qquad \Gamma' \mid e : B \vdash \Delta'}{\Gamma, \Gamma' \mid v \cdot e : A \to B \vdash \Delta, \Delta'}\ (\to \vdash)$$

$$\frac{\Gamma \vdash v : A \to B \mid \Delta \qquad \Gamma' \vdash v' : A \mid \Delta'}{\Gamma, \Gamma' \vdash v\, v' : B \mid \Delta, \Delta'}$$

Étudions deux cas particuliers.

Cas $N \to M$. Cela correspond à l'appel par nom. On a, pour E une valeur positive :

$$\begin{aligned}
\langle v\, v' \,\|\, E\rangle &\to_h \langle v \,\|\, v' \cdot E\rangle \\
\langle \lambda\alpha.v \,\|\, v' \cdot E\rangle &\to_h^2 \langle v\,[v'/\alpha] \,\|\, E\rangle
\end{aligned}$$

On retrouve les règles de réduction d'une machine de Krivine [Kri04], dont les *piles* sont des environnements-valeurs ; ou encore les règles du $\bar\lambda\mu\tilde\mu$ en appel par nom [CH00].

Cas $P \to Q$. Cela correspond-il à l'appel par valeur ? On a :

$$\begin{aligned}
\langle v\, v' \,\|\, e\rangle &\to_h \langle v \,\|\, v' \cdot e\rangle \\
\langle \lambda x.v \,\|\, v' \cdot e\rangle &\to^6 \langle v' \,\|\, \mu x.\langle v \,\|\, E\rangle\rangle
\end{aligned}$$

Cela ressemble aux règles du $\bar\lambda\mu\tilde\mu$ en appel par valeur, mais c'est par accident ! En effet on a de façon plus générale :

$$\langle v \,\|\, v' \cdot e\rangle \to^4 \langle v' \,\|\, \mu x.\langle v \,\|\, x \cdot E\rangle\rangle$$

Cela est dû au fait que $\lambda x.v$ est négatif au lieu d'être de la polarité de sa sortie. Un $\lambda x.v$ positif est demandé par des pratiques de programmation courantes telles que :

```
(*OCaml *)
let f =
  let r = ref []
  in function -> ...
```

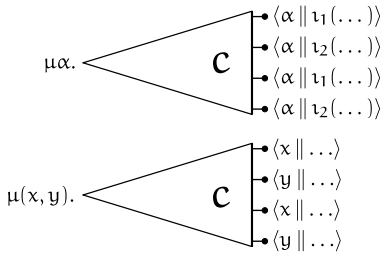
qui demandent que la fonction soit évaluée de manière stricte. C'est en prenant en prenant $\downarrow(P^\perp \parr Q)$ que l'on retrouve l'implication en appel par valeur à la Curien-Herbelin.

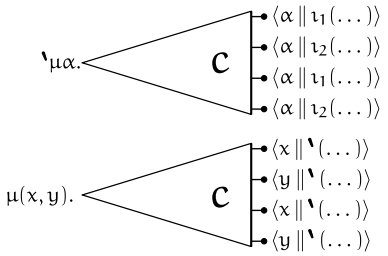
3.7 D'autres protocoles sont-ils possibles ?

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* [Her05], qui est connue pour être implémentée à travers le langage Haskell. L'étude de cette stratégie dépasse la portée de cet article.

Dissipons également une objection qui serait l'apparence arbitraire du choix de privilégier les positifs plutôt que les négatifs. Il revient au même du point de vue calculatoire de privilégier systématiquement la réduction négative sur la réduction positive. En effet, procéder ainsi force artificiellement les connecteurs paresseux à être stricts et vice-versa — il s'agit du même système mais dans une écriture franchement *ad hoc*.

La syntaxe est docile En effet, comparons la géométrie de deux termes similaires, dont la seule différence est que l'un est formulé avec des constructeur $\oplus$ et l'autre avec un constructeur $\parr$:





On remarque tout d'abord que dans le cas du $\oplus$, les constructeurs sont aux feuilles, tandis que dans le cas du $\parr$, le constructeur est à la racine. La première conséquence est que le terme formulé avec $\oplus$ n'a besoin d'évaluer son contre-terme que lorsque celui-ci 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 terme formulé avec $\parr$, quant à lui, doit évaluer son contre-terme jusqu'à ce que celui-ci soit de la forme (v, v'), puisque le constructeur est en tête. Il a donc un caractère paresseux, puisqu'il contraint son dual $\otimes$ à adopter un comportement strict.

Ces remarques étant faites, il est évident qu'une stratégie qui donne priorité au lieur μx plutôt qu'au lieur $\mu\alpha$ force $\oplus$ à se comporter en $\parr$. C'est la voie empruntée dans [Wad03] pour définir les stratégies de réduction du Calcul Dual. Pour parfaire un tel *jeu de mots* par lequel on emploie la même notation pour dénoter tour à tour la paire stricte et la paire paresseuse, l'auteur rejette la réversibilité du & et attribue à ce dernier les réductions du $\otimes$.

Interlude : Enter quote Le caractère *paresseux* ou *strict* est donc lié à la géométrie du constructeur : le $\mu(x, y)$ est *bloquant*[5] alors que le $\mu\alpha$ ne l'est pas. Être bloquant est la qualité essentielle du quote $`(\cdot)$. Avec celui-ci, on parvient ainsi à donner à $\oplus$ et à $\parr$ un comportement commun :

[5] Il 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...

que l'on fait réagir respectivement avec des programmes produisant du $\mu`(x).\langle x \parallel (t \mid t')\rangle$ et du $(\mu`(x).\langle x \parallel t\rangle, \mu`(x).\langle x \parallel t'\rangle)$. Le fait que les deux nouveaux termes sont isomorphes en termes de parties bloquantes devrait rendre évident le fait qu'ils ont le même comportement calculatoire.

Cette remarque donne une nouvelle lecture des équations bien connues :

$$\begin{aligned} ?(A^\perp \oplus B^\perp) &\simeq ?A^\perp \parr ?B^\perp \\ !(A \,\&\, B) &\simeq !A \otimes !B \end{aligned}$$

On a mis en évidence, du point de vue procédural, le *caractère exponentiel du* $`(\cdot)$ — en dehors toute règle de typage.

4 Réalisabilité

On définit un outil observationnel pour l'étude du système L focalisant pur basé sur la réalisabilité classique de Krivine pour le λ-calcul [Kri04].

4.1 Orthogonal

Definition 24. Une partie $\perp\!\!\!\perp$ de $\mathcal{C}^0$ est dite *saturée* si :

$$c \to_h c', c' \in \perp\!\!\!\perp \Longrightarrow c \in \perp\!\!\!\perp$$

On travaillera donc essentiellement sur des termes clos. Remarquer que ce n'est pas une limitation : tout terme peut être vu comme un terme clos si l'on décide que ses variables libres sont des constantes.

Dans la suite, $\perp\!\!\!\perp$ est une partie saturée.

Definition 25. Soit C une partie de $\mathcal{C}^0$. On pose :

$$\mathcal{S}at(C) = \left\{ c \in \mathcal{C}^0 \;\middle|\; \exists c' \in C, c \to_h^* c' \right\}$$

la *clôture saturée* de C.

Definition 26.

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

$$T^\perp = \left\{ t_- \in \mathcal{T}_-^0 \;\middle|\; \forall t_+ \in T, \langle t_+ \parallel t_- \rangle \in \perp\!\!\!\perp \right\}$$

2. Soit $T \in \mathcal{P}(\mathcal{T}_-^0)$. On pose :

$$T^\perp = \left\{ t_+ \in \mathcal{T}_+^0 \;\middle|\; \forall t_- \in T, \langle t_+ \parallel t_- \rangle \in \perp\!\!\!\perp \right\}$$

3. Un *comportement* est une partie de $\mathcal{T}_+^0$ ou de $\mathcal{T}_-^0$ de la forme $T^\perp$.

Il y a un conflit de notation ci-dessus, selon que l'on considère $\emptyset$ comme étant une partie de $\mathcal{T}_-^0$ ou bien comme étant une partie de $\mathcal{T}_+^0$. Le contexte permettra alors de décider si l'on a $\emptyset^\perp = \mathcal{T}_+^0$ (premier cas) ou $\emptyset^\perp = \mathcal{T}_-^0$ (second cas).

Proposition 27 (Propriétés de base de l'orthogonal)**.** Soient T et U deux parties de $\mathcal{T}_+^0$ ou de $\mathcal{T}_-^0$.

1. On a $T \subset T^{\perp\perp}$;
2. Si $T \subset U$ alors $U^\perp \subset T^\perp$;
3. On a $T^{\perp\perp\perp} = T^\perp$;
4. T est un comportement si et seulement si $T = T^{\perp\perp}$;
5. Si $\mathcal{U}$ est un ensemble de parties de $\mathcal{T}_+^0$ ou de $\mathcal{T}_-^0$, alors on a :

$$\left(\bigcup \mathcal{U}\right)^\perp = \bigcap \left\{ T^\perp \;\middle|\; T \in \mathcal{U} \right\}$$

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

Démonstration. Ce sont les résultats usuels d'une telle construction. □

4.2 Réalisabilité

Definition 28 (Paramètres)**.** On définit $\Pi = \mathcal{P}(\mathcal{T}_+^0 \cap \mathbb{V})$ l'ensemble des *paramètres.*

Ces paramètres seront dénotés par R, S et on étend le langage des formules de la façon suivante : lorsque R est un paramètre, alors R est une formule atomique positive et $R^\perp$ est une formule atomique négative.

Definition 29. Soient T et U deux parties de $\mathcal{T}_+^0$ ou de $\mathcal{T}_-^0$. On pose :

$$\begin{aligned} T \times U &= \{ (t,u) \mid t \in T, u \in U \} \\ T + U &= \{ \iota_1(t) \mid t \in T \} \cup \{ \iota_2(u) \mid u \in U \} \\ \downarrow T &= \{ \{u\} \mid u \in T \} \\ !T &= \left\{ \mu`(\kappa).c \;\middle|\; V \in T^\perp{}_{\mathbb{V}} \Rightarrow c\,[V/\kappa] \in \perp\!\!\!\perp \right\} \end{aligned}$$

Definition 30. À toute formule close positive P on associe un comportement $|P| \in \mathcal{P}(\mathcal{T}_+^0)$ et à toute formule close négative N on associe un comportement $|N| \in \mathcal{P}(\mathcal{T}_-^0)$. Pour tout terme t, on dit alors que t *réalise* A, et on écrit $t \Vdash A$, si $t \in |A|$. On procède par récurrence sur la formule considérée :

$$|R^\perp| = R^\perp$$
$$|R| = R^{\perp\perp}$$
$$|A \parr B| = (|A^\perp| \times |B^\perp|)^\perp$$
$$|A \otimes B| = (|A| \times |B|)^{\perp\perp}$$
$$|A \,\&\, B| = (|A^\perp| + |B^\perp|)^\perp$$
$$|A \oplus B| = (|A| + |B|)^{\perp\perp}$$
$$|\bot| = \{()\}^\perp$$
$$|\mathbf{1}| = \{()\}^{\perp\perp}$$
$$|\top| = \emptyset^\perp$$
$$|\mathbf{0}| = \emptyset^{\perp\perp}$$
$$|?A| = (!|A^\perp|)^\perp$$
$$|!A| = (!|A|)^{\perp\perp}$$
$$|\forall X\, A| = \left(\bigcup_{R \in \Pi} \downarrow \left| A^\perp\,[R/X] \right| \right)^\perp$$
$$|\exists X\, A| = \left(\bigcup_{R \in \Pi} \downarrow \left| A\,[R/X] \right| \right)^{\perp\perp}$$

Proposition 31. Pour toute formule A, on a :

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

Démonstration. Immédiat par définition. □

Corollary 32. Soit t un terme clos et A une formule close. On a $t \Vdash A$ si et seulement si $\forall u\, (u \Vdash A^\perp \Rightarrow \langle t \parallel u \rangle \in \perp\!\!\!\perp)$.

Remark 33. Une formule positive P quelconque est réalisée dès lors que $\perp\!\!\!\perp \neq \emptyset$. En effet, si $c \in \perp\!\!\!\perp$, on a $\mu\_.c \Vdash P$, où $\_$ est une variable négative.

4.3 Théorème de génération

Definition 34. Soient T un comportement et U une partie de T. On dit que U est un générateur de (ou engendre) T si $T = U^{\perp\perp}$.

Definition 35. Soit T un comportement. L'ensemble $T_{\mathbb{V}}$ des *valeurs de* T est l'intersection de T et de l'ensemble des valeurs $\mathbb{V}$.

Theorem 36 (Génération)**.** *Si A est une formule close, alors $|A|$ est engendré par l'ensemble de ses valeurs.*

La preuve nécessite les lemmes qui suivent. Comme on le verra en section 5.2, un comportement quelconque n'est pas nécessairement engendré par l'ensemble de ses valeurs.

Lemma 37. Soit T une partie de $\mathcal{T}_+^0$ (resp. $\mathcal{T}_-^0$) et $c \in \mathcal{C}$ avec $\mathcal{FV}(c) = \{\kappa\}$ avec κ une variable positive (resp. négative). Si pour tout $V \in T_{\mathbb{V}}$, on a $c[V/\kappa] \in \perp\!\!\!\perp$, alors $\mu\kappa.c \in T_{\mathbb{V}}{}^{\perp}$.

Démonstration. Soit $V \in T_{\mathbb{V}}$. On a $\langle V \parallel \mu\kappa.c\rangle \to_\mu c[V/\kappa] \in \perp\!\!\!\perp$, donc $\langle V \parallel \mu\kappa.c\rangle \in \perp\!\!\!\perp$ par saturation de $\perp\!\!\!\perp$. D'où le résultat. □

Lemma 38. Si $U \subset U' \subset T$ et U engendre T, alors U' engendre T. *A fortiori*, si un comportement est engendré par un ensemble de valeurs, alors il est engendré par l'ensemble de ses valeurs.

Démonstration. Trivial. □

Lemma 39. Si T est un comportement, alors $T_{\mathbb{V}}{}^{\perp\perp} \cap \mathbb{V} = T_{\mathbb{V}}$.

Démonstration. ($\supset$) Trivial. ($\subset$) T étant un comportement, le membre de droite s'écrit $T^{\perp\perp} \cap \mathbb{V}$. Or $T_{\mathbb{V}}{}^{\perp\perp} \subset T^{\perp\perp}$, d'où le résultat. □

Lemma 40. Soit T et U deux comportements. Les propriétés suivantes sont vérifiées :

1. $T_{\mathbb{V}}{}^{\perp\perp} \times U_{\mathbb{V}}{}^{\perp\perp} \subset (T \times U)_{\mathbb{V}}{}^{\perp\perp}$;
2. $T_{\mathbb{V}}{}^{\perp\perp} + U_{\mathbb{V}}{}^{\perp\perp} \subset (T + U)_{\mathbb{V}}{}^{\perp\perp}$;
3. $\downarrow(T_{\mathbb{V}}{}^{\perp\perp}) \subset (\downarrow T_{\mathbb{V}})^{\perp\perp}$.

Démonstration. (1) Par définition, on a $(T \times U)_{\mathbb{V}} = T_{\mathbb{V}} \times U_{\mathbb{V}}$. Soit $t \in T_{\mathbb{V}}{}^{\perp\perp}$ et $u \in U_{\mathbb{V}}{}^{\perp\perp}$; soit $\nu \in (T_{\mathbb{V}} \times U_{\mathbb{V}})^{\perp}$. Si $t, u \in \mathbb{V}$, alors $(t,u) \in T_{\mathbb{V}} \times U_{\mathbb{V}}$ par le lemme 39, d'où $\langle (t,u) \parallel \nu\rangle \in \perp\!\!\!\perp$. Sinon, on a :

$$\langle (t,u) \parallel \nu\rangle \to_h \left\langle t \,\middle\|\, \mu\kappa.\left\langle u \,\middle\|\, \mu\kappa'.\langle (\kappa,\kappa') \parallel \nu\rangle\right\rangle\right\rangle$$

Par saturation de $\perp\!\!\!\perp$, il suffit donc de montrer $\left\langle t \,\middle\|\, \mu\kappa.\langle u \parallel \mu\kappa'.\langle (\kappa,\kappa') \parallel \nu\rangle\rangle\right\rangle \in \perp\!\!\!\perp$. Mais pour tous $t' \in T_{\mathbb{V}}$ et $u' \in U_{\mathbb{V}}$, par définition de ν, on a $\langle (t',u') \parallel \nu\rangle \in \perp\!\!\!\perp$. Donc $\mu\kappa'.\langle (u',\kappa') \parallel \nu\rangle \in U_{\mathbb{V}}{}^{\perp}$ par le lemme 37. Donc aussi $\left\langle u \,\middle\|\, \mu\kappa'.\langle (t',\kappa') \parallel \nu\rangle\right\rangle \in \perp\!\!\!\perp$ par hypothèse sur u. Donc aussi $\mu\kappa.\left\langle u \,\middle\|\, \mu\kappa'.\langle (\kappa,\kappa') \parallel \nu\rangle\right\rangle \in T_{\mathbb{V}}{}^{\perp}$ par le lemme 37. D'où le résultat par hypothèse sur t. (2) et (3) : même raisonnement. □

Démonstration (Théorème). 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 par convention. Si A est un paramètre $R \in \Pi$, alors $|A| = R^{\perp\perp}$. R étant un ensemble de valeurs, $|A|$ est engendré par l'ensemble de ses valeurs. Cas où $A = B \otimes C$: par hypothèse de récurrence, $|B| \times |C|$ est égal à ${|B|_{\mathbb{V}}}^{\perp\perp} \times {|C|_{\mathbb{V}}}^{\perp\perp}$, lequel est inclus dans $(|B| \times |C|)_{\mathbb{V}}{}^{\perp\perp}$ par le lemme 40. Donc $|A|$ est généré par $(|B| \times |C|)_{\mathbb{V}}$. Cas où $A = B \oplus C$: même preuve en remplaçant $\times$ par $+$. Cas où $A = !B$: $|A|$ est égal à $(!|B|)^{\perp\perp}$ avec $!|B|$ un ensemble de valeurs ; $|A|$ est donc engendré par ses valeurs. Cas où $A = \mathbf{1}$: A est engendré par $\{()\}$; il est donc engendré par ses valeurs. Cas où $A = \mathbf{0}$: A est engendré par $\emptyset$; il est donc engendré par ses valeurs. Cas où $A = \exists X\, B$: On a par hypothèse de récurrence, pour tout $R \in \Pi$, ${|B[R/X]|_{\mathbb{V}}}^{\perp} \subset |B[R/X]|^{\perp}$, d'où l'on déduit à l'aide du le lemme 40 $(\downarrow|B[R/X]|)_{\mathbb{V}}{}^{\perp} \subset (\downarrow|B[R/X]|)^{\perp}$. Donc aussi :

$$\bigcap_{R\in\Pi} (\downarrow|B[R/X]|)_{\mathbb{V}}{}^{\perp} \subset \bigcap_{R\in\Pi} (\downarrow|B[R/X]|)^{\perp}$$

Par propriété de base de l'orthogonal (27), on a donc que $|\exists X\, B|$ est engendré par $\bigcup_{R\in\Pi} (\downarrow|B[R/X]|)_{\mathbb{V}}$ qui est bien un ensemble de valeurs. □

Corollary 41. Soit A une formule close et $c \in \mathcal{C}$ avec $\mathcal{FV}(c) = \{\kappa\}$ avec κ une variable de même polarité que A. Pour montrer $\mu\kappa.c \Vdash A^{\perp}$, il suffit de montrer que $V \in |A|_{\mathbb{V}}$ implique $c[V/\kappa] \in \perp\!\!\!\perp$.

Démonstration. Découle du lemme 37 et du théorème de génération. □

Corollary 42 (Substitution)**.** Soit A une formule avec $\mathcal{FV}(A)$ de la forme $\{X\}$ et P une formule close positive. On a :

$$\left|A\,[P/X]\right| = \left|A\,[|P|_{\mathbb{V}}/X]\right|$$

Démonstration. Par récurrence aisée sur la taille de la formule A. Les cas importants sont pour A un atome. Si $A = X$, alors on a $\left|A\left[|P|_{\mathbb{V}}/X\right]\right| = \||P|_{\mathbb{V}}\| = |P|_{\mathbb{V}}{}^{\perp\perp}$, et similairement pour $A = X^{\perp}$, le théorème de génération permettant de conclure. □

4.4 Lemme d'adéquation

Le lemme d'adéquation affirme que les systèmes de preuves sont un moyen d'obtenir des termes de comportements donnés par leurs types.

Theorem 43 (Lemme d'adéquation, $\mathsf{LK}_{\mathrm{pol}}$)**.** *Soit* c *une commande (respectivement* t *un terme) typable dans* $\mathsf{LK}_{\mathrm{pol}}$*, de type* $\vdash \kappa_1 : A_1, \dots, \kappa_n : A_n$ *(resp.* $\vdash t : B \mid \kappa_1 : A_1, \dots, \kappa_n : A_n$*) avec* $A_1, \dots, A_n$ *(resp. et* B*) des formules closes. Alors pour tous termes clos* $u_1, \dots, u_n$*, si* $u_1 \Vdash A_1^{\perp}, \dots, u_n \Vdash A_n^{\perp}$*, alors* $c[u_1/\kappa_1, \dots, u_n/\kappa_n] \in \perp\!\!\!\perp$ *(resp.* $t[u_1/\kappa_1, \dots, u_n/\kappa_n] \Vdash B$*).*

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

Lemma 44. Soit c une commande (respectivement t un terme) typable dans $\mathsf{LK}_{\mathrm{pol}}$, de type $\vdash \Gamma$ (resp. de type $\vdash t : B \mid \Gamma$) où $\Gamma = \kappa_1 : A_1, \dots, \kappa_n : A_n$. Ces formules ont $X_1, \dots, X_m$ pour variables libres. Soit des paramètres $R_1, \dots, R_m \in \Pi$ et des termes clos $u_1, \dots, u_n$. On note $\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right]$ la substitution $[u_1/\kappa_1, \dots, u_n/\kappa_n]$ et $\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]$ la substitution $[R_1/X_1, \dots, R_m/X_m]$; on pose pour tout $i \leq n$ la formule close $A'_i = A_i\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]$. Si $u_1 \Vdash A'^{\perp}_1, \dots, u_n \Vdash A'^{\perp}_n$, alors $c\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \in \perp\!\!\!\perp$ (resp. $t\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \Vdash B'$ où $B' = B\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]$).

Démonstration. Par récurrence sur la dérivation de c et de t. (Axiome) Trivial. (Activation) $\vdash \mu\kappa.c : B \mid \Gamma$ provient de $c : (\vdash \kappa : B \mid \Gamma)$. Pour montrer $(\mu\kappa.c)\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \Vdash B'$, il suffit de montrer, par le corollaire 41 que pour tout $V \in |B^{\perp}|_{\mathbb{V}}$, on a $c\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right][V/\kappa] \in \perp\!\!\!\perp$: cela est fourni par l'hypothèse de récurrence. (Coupure) $\langle t \,\|\, t'\rangle : (\vdash \Gamma, \Gamma')$ provient de $\vdash t : B \mid \Gamma$ et de $\vdash t' : B^{\perp} \mid \Gamma'$, avec $\Gamma = \{\kappa_j : A_j \mid j \leq i\}$ et $\Gamma' = \{\kappa_j : A_j \mid j > i\}$ étant donné $0 \leq i \leq n$. On a par hypothèse de récurrence $t\left[\overrightarrow{u}_{j\leq i}/\overrightarrow{\kappa}_{j\leq i}\right] \Vdash B'$ et $t'\left[\overrightarrow{u}_{j>i}/\overrightarrow{\kappa}_{j>i}\right] \Vdash B'^{\perp}$. Puisque $\left|B^{\perp}\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]\right| = \left|B\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]\right|^{\perp}$, on a bien $\langle t \,\|\, t'\rangle\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \in \perp\!\!\!\perp$. (Affaiblissement, contraction) Trivial. (Tenseur) $\vdash (t, u) : B \otimes C \mid \Gamma, \Gamma'$ provient de $\vdash t : B \mid \Gamma$ et de $\vdash u : C \mid \Gamma'$. Le résultat suit de ce que $|B'| \times |C'| \subset |B' \otimes C'|$. (Plus) Cas similaire au tenseur. (Par) $\vdash \mu(\kappa, \kappa').c : B \parr C \mid \Gamma$ provient de $c : (\vdash \kappa : B, \kappa' : C, \Gamma)$. Soit $(V, V') \in |B'^{\perp}|_{\mathbb{V}} \times |C'^{\perp}|_{\mathbb{V}}$. On pose $c' = c\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right]$. On a $\langle (V, V') \,\|\, \mu(\kappa, \kappa').c'\rangle \to_h c'[V, V'/\kappa, \kappa'] \in \perp\!\!\!\perp$ par hypothèse de recurrence. Donc $\mu(\kappa, \kappa').c' \in (|B'^{\perp}| \times |C'^{\perp}|)_{\mathbb{V}}{}^{\perp}$. C'est bien inclus dans $(|B'^{\perp}| \times |C'^{\perp}|)^{\perp}$ par le théorème de génération et le lemme 40. (Avec) Cas similaire au par. (Extraction) $\vdash \{t\} : \exists X\, B \mid \Gamma$ provient de $\vdash t : B[P/X] \mid \Gamma$, avec X supposé distinct des $X_1, \dots, X_m$. On pose $P' = P\left[\overrightarrow{R}_j\middle/\overrightarrow{X}_j\right]$. Par le corollaire de substitution 42, on a $\left|B'[P'/X]\right| = \left|B'\left[|P'|_{\mathbb{V}}/X\right]\right|$. Par hypothèse de récurrence, on a $t\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \Vdash B'[P'/X]$. Donc $\{t\}\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \in \downarrow\left|B'\left[|P'|_{\mathbb{V}}/X\right]\right|$. D'où $\{t\}\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \Vdash \exists X\, B'$. (Généralisation) $\vdash \mu\{\kappa\}.c : \forall X\, B \mid \Gamma$ provient de $c : (\vdash \kappa : B \mid \Gamma)$, avec X distinct des $X_1, \dots, X_m$. Par propriété de base de l'orthogonal, on a $|\forall X\, B'| = \bigcap_{R\in\Pi}\left(\downarrow\left|B'[R/X]\right|\right)^{\perp}$. Soit $R \in \Pi$ et $\{V\} \in \downarrow\left|B'[R/X]\right|_{\mathbb{V}}$. On a $\langle\{V\} \,\|\, \mu\{\kappa\}.c'\rangle \to_h c'[V/\kappa] \in \perp\!\!\!\perp$ par hypothèse de récurrence, donc $\mu\{\kappa\}.c' \in \left(\downarrow\left|B'[R/X]\right|\right)_{\mathbb{V}}{}^{\perp}$. C'est bien inclus dans $\left(\downarrow\left|B'[R/X]\right|\right)^{\perp}$ par le théorème de génération et le lemme 40. (**1**, $\perp$ et $\top$) Trivial.

□

Remark 45. Le lemme d'adéquation reste valable si l'on substitue « LL » pour « $\mathsf{LK}_{\mathrm{pol}}$ ». Il faut alors étendre la preuve avec les cas qui suivent. Le reste de la preuve est la même — les contractions et les affaiblissements sont en particulier gratuits. Cela montre que la réalisabilité n'est pas impliquée vis-à-vis de notions procédurales telle la linéarité.

Démonstration (Lemme d'adéquation, LL*).* (Déréliction) $\vdash \text{`}(t) : ?B \mid \kappa_1 : A_1, \dots, \kappa_n : A_n$ provient de $\vdash t : B \mid \kappa_1 : A_1, \dots, \kappa_n : A_n$. On pose $t' = t\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right]$. Soit $\mu\text{`}(\kappa).c \in\, !\left|B'^{\perp}\right|$. On a donc par définition $V \in |B'| \Rightarrow c[V/\kappa] \in \perp\!\!\!\perp$. Si t' est une valeur, alors $\langle\mu\text{`}(\kappa).c \,\|\, \text{`}(t')\rangle \to_h c[t'/\kappa] \in \perp\!\!\!\perp$ puisque par hypothèse de récurrence $t' \in |B'|$, ce qui montre le résultat. Sinon, on a $\langle\mu\text{`}(\kappa).c \,\|\, \text{`}(t')\rangle \to_h \left\langle t' \middle\| \mu\kappa'.\langle\mu\text{`}(\kappa).c \,\|\, \text{`}(\kappa')\rangle\right\rangle$ avec $\mu\kappa'.\langle\mu\text{`}(\kappa).c \,\|\, \text{`}(\kappa')\rangle \in \left|B'^{\perp}\right|$, d'où le résultat avec $t' \in |B'|$. (Promotion) $\vdash \mu\text{`}(\kappa).c : !B \mid \kappa_1 : ?A_1, \dots, \kappa_n : ?A_n$ provient de $c : (\vdash \kappa : B, \kappa_1 : ?A_1, \dots, \kappa_n : ?A_n)$. On a par récurrence, pour tout $t \in |B'^{\perp}|$, $c\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right][t/\kappa] \in \perp\!\!\!\perp$. D'où $\mu\text{`}(\kappa).c\left[\overrightarrow{u}_i/\overrightarrow{\kappa}_i\right] \in\, !|B'|$. □

4.5 Complétude interne

Le titre renvoie à la complétude interne des connecteurs de la Ludique [Gir07]. La réalisabilité admet ici une notion analogue.

Definition 46. Un ensemble R de valeurs closes de même polarité est *complet* si l'une des deux propriétés équivalentes suivantes est vérifiée :

1. $R^{\perp\perp_{\mathbb{V}}} = R$.
2. R est de la forme $T_{\mathbb{V}}$ avec T un comportement.

L'équivalence provient du lemme 39.

Proposition 47. Soit R et S complets. Les propriétés suivantes sont vérifiées :

1. $\left(R^{\perp\perp} \times S^{\perp\perp}\right)^{\perp\perp} = (R \times S)^{\perp\perp}$;
2. $\left(R^{\perp\perp} + S^{\perp\perp}\right)^{\perp\perp} = (R + S)^{\perp\perp}$;
3. $\downarrow(R^{\perp\perp}) = (\downarrow R)^{\perp\perp}$.

Démonstration. Il s'agit d'une reformulation du lemme 40. □

Il est aisé de trouver des contre-exemples *ad hoc* à la complétude des connecteurs dans le cas d'une observation quelconque. Dès lors, il s'agit de trouver un ensemble *raisonnable* de contraintes sur l'observation pour obtenir la complétude interne de nos connecteurs.

Proposition 48 (Complétude interne)**.** Soit $\perp\!\!\!\perp$ une observation non-vide, qui ne contient pas de commande mal-formée et qui est stable par la réduction $\rightarrow_\beta$. On a alors :

1. $\emptyset$ (en tant qu'ensemble de termes positifs) et $\{()\}$ sont complets ;
2. Pour tous R et S complets, les ensembles $R \times S$, $R + S$ et $\downarrow R$ sont complets.
3. Pour T un ensemble de termes de même polarité tel que $T^{\perp}_{\mathbb{V}} \neq \emptyset$, $!T$ est complet.

Démonstration. Soit $\perp\!\!\!\perp$ une telle observation. On pose c_0 un élement de $\perp\!\!\!\perp$. (1) Soit $V \in \emptyset^{\perp\perp_{\mathbb{V}}}$. On a pour tout $t \in \mathcal{T}^0_-$, $\langle V \parallel t\rangle \in \perp\!\!\!\perp$. C'est impossible, car parmi ces commandes il en existe de mal-formées. Donc $\emptyset^{\perp\perp_{\mathbb{V}}} = \emptyset$. Soit $V \in \{()\}^{\perp\perp_{\mathbb{V}}}$. On vérifie $\mu().c_0 \in \{()\}^{\perp}$. On a donc $\langle V \parallel \mu().c_0\rangle \in \perp\!\!\!\perp$, ce dernier n'est donc pas mal-formé et on a bien $V = ()$ (on obsevera ici et dans le reste de la preuve l'importance de travailler sur des termes clos). (2) Soit R et S complets. Soit $V \in (R \times S)^{\perp\perp_{\mathbb{V}}}$. On a $\mu(\_,\_).c_0 \in (R \times S)^{\perp}$, donc $\langle V \parallel \mu(\_,\_).c_0\rangle \in \perp\!\!\!\perp$ et cette dernière n'est donc pas mal-formée. On a alors V de la forme (V_1, V_2) avec V_1 et V_2 respectivement des mêmes polarités que R et S. Soit alors $t \in R^{\perp}$. On a $\mu(\kappa,\_).\langle \kappa \parallel t\rangle \in (R \times S)^{\perp}$. Donc $\left\langle (V_1, V_2) \,\middle\|\, \mu(\kappa,\_).\langle \kappa \parallel t\rangle \right\rangle \in \perp\!\!\!\perp$. Comme $\perp\!\!\!\perp$ est stable par la réduction $\rightarrow_\beta$, on a $\langle V_1 \parallel t\rangle \in \perp\!\!\!\perp$. Donc $V_1 \in R^{\perp\perp}$. D'où $V_1 \in R$ par la complétude de R. On montre de façon similaire $V_2 \in S$, donc on a bien $V \in R \times S$. On montre les complétudes de $R + S$ et de $\downarrow R$ de façon similaire. (3) Soit T un ensemble de termes de même polarité tel qu'il existe $V_0 \in T^{\perp}_{\mathbb{V}}$. Soit $V \in (!T)^{\perp\perp_{\mathbb{V}}}$. On a $`(V_0) \in (!T)^{\perp}$. D'où $\langle V \parallel `(V_0)\rangle \in \perp\!\!\!\perp$, donc cette commande est bien formée et V est donc de la forme $\mu`(\kappa).c$. Soit alors $V' \in T^{\perp}_{\mathbb{V}}$. On a $`(V') \in (!T)^{\perp}$, donc $\langle \mu`(\kappa).c \parallel `(V')\rangle \in \perp\!\!\!\perp$. $\perp\!\!\!\perp$ étant stable par la réduction $\rightarrow_\beta$, on a $c\left[V'/\kappa\right] \in \perp\!\!\!\perp$. Ceci montre $V \in !T$. □

Corollary 49. Lorsque R et S sont complets et que $\perp\!\!\!\perp$ répond aux critères de la proposition précédente, on a $|R \otimes S|_{\mathbb{V}} = R \times S$. En particulier, $|A \otimes B|_{\mathbb{V}} = |A|_{\mathbb{V}} \times |B|_{\mathbb{V}}$. (Et similairement pour les connecteurs $+$ et $\downarrow$.)

Démonstration. La première égalité est une conséquence des deux propositions précédentes. La seconde de la première et du corollaire de substitution 42. □

Proposition 50. Soit $C \in \mathcal{P}(\mathcal{C}^0)$. Si C est non-vide, stable par la réduction $\rightarrow_\beta$ et ne contient pas de terme mal-formé, alors de même pour l'observation $Sat(C)$.

Démonstration. Les termes mal-formés de $Sat(C)$ sont ceux de C, car ceux-ci par définition ne se réduisent pas. S'ensuit le résultat de manière évidente. □

5 Applications

Dans ce qui suit, $\vdash$ désigne indifféremment la typabilité dans $\mathsf{LK}_{\mathrm{pol}}$ ou dans LL.

5.1 Propriétés issues de la réalisabilité

Le theorème de géneration et le lemme d'adéquation possèdent les corollaires suivants :

Corollary 51 (Normalisation de tête)**.** Soit P une formule positive close et $t \in \mathcal{T}_+^0$ tel que $\vdash t : P$. Alors il existe une valeur $V \in \mathcal{T}_+^0$ tel que :

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

Démonstration. On pose $\perp\!\!\!\perp = \mathcal{S}at\left\{ \langle V \parallel tp \rangle \;\middle|\; V \text{ valeur close positive} \right\}$. On a pour tout $V \in |P|_{\mathbb{V}}$, $\langle V \parallel tp \rangle \in \perp\!\!\!\perp$, c'est-à-dire $tp \in |P|_{\mathbb{V}}{}^{\perp}$, et donc $tp \Vdash P^{\perp}$ par le théorème de génération 36. Or, par le lemme d'adéquation, on a $t \Vdash P$. D'où $\langle t \parallel tp \rangle \in \perp\!\!\!\perp$, ce qu'il faut démontrer. □

Les propriétés comme celle de la disjonction résultent habituellement d'un théorème d'élimination des coupures et de la réduction du sujet. L'exemple suivant montre que le lemme d'adéquation peut parfois remplacer ces deux derniers.

Example 52 (Propriété de la disjonction)**.** Soit une formule de la forme $A_1 \oplus A_2$ et $t \in \mathcal{T}_+^0$ tel que $\vdash t : A_1 \oplus A_2$. Alors il existe $i \in \{1,2\}$ et une valeur close V de la même polarité que A_i tels que :

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

Démonstration. On pose C l'ensemble des $\langle \iota_i(V) \parallel tp \rangle$ avec $i \in \{1,2\}$ et V une valeur close de la même polarité que A_i. On pose $\perp\!\!\!\perp = \mathcal{S}at(C)$. On a pour tout $V \in |A_i|_{\mathbb{V}}$, $\langle \iota_i(V) \parallel tp \rangle \in \perp\!\!\!\perp$, donc $tp \in (|A_1|_{\mathbb{V}} + |A_2|_{\mathbb{V}})^{\perp}$. Par la proposition 47 et le théorème de génération, on a donc $tp \Vdash A_1{}^{\perp} \,\&\, A_2{}^{\perp}$. Or, par le lemme d'adéquation, on a $t \Vdash A_1 \oplus A_2$. D'où $\langle t \parallel tp \rangle \in \perp\!\!\!\perp$, ce qu'il faut démontrer. □

Remark 53. Cette méthode se généralise pour un type positif quelconque. On obtient ainsi la forme générale du résultat en décomposant le type jusqu'aux exponentielles et jusqu'aux changements de polarité. Cela donne donc une propriété de focalisation pour LL. Cette propriété est aussi vraie dans $\mathsf{LK}_{\mathrm{pol}}$, mais remarquons qu'elle est ternie par le fait que la valeur obtenue peut désormais dépendre du contre-terme (en l'occurrence tp) si une contraction sur une variable négative intervient lors de la réduction.

Une autre application de la réalisabilité est l'obtention de résultats de paramétricité :

Example 54 (Paramétricité)**.** Soit t un terme typable de type $\vdash t : \forall X(X \otimes X \to X \otimes X)$ et V_1 et V_2 deux valeurs positives. On a $\langle t \parallel \{((V_1, V_2), tp)\} \rangle \to_h^* \langle (V_i, V_j) \parallel tp \rangle$ avec $i, j \in \{1,2\}$. En effet, posons $\perp\!\!\!\perp = \mathcal{S}at\left\{ \langle (V_i, V_j) \parallel tp \rangle \;\middle|\; i,j \in \{1,2\} \right\}$ et $R = \{V_1, V_2\} \in \Pi$. On peut dériver $\langle t \parallel \{((V_1, V_2), \alpha)\} \rangle : (\vdash \alpha : R \otimes R)$. Or $tp \Vdash R^{\perp} \parr R^{\perp}$. Par le lemme d'adéquation, on a donc $\langle t \parallel \{((V_1, V_2), tp)\} \rangle \in \perp\!\!\!\perp$.

Remark 55. Dans LL, on ne doit obtenir ni $\langle (V_1, V_1) \parallel tp \rangle$, ni $\langle (V_2, V_2) \parallel tp \rangle$. Cependant, en conséquence de la remarque 45, on sait que ce n'est pas l'outil développé ici qui va permettre montrer ce résultat.

5.2 Quantification, décalage et restriction aux valeurs

Nos règles de typage introduisent un décalage lors de la dérivation des quantifications du second ordre, afin de donner respectivement aux connecteurs $\exists$ et $\forall$ les polarités positive et négative.

L'impossibilité d'un $\forall$ positif est remarquée par Girard lorsque celui-ci développe une sémantique dénotationnelle pour la logique classique [Gir91]. Plus tard, il montre cependant qu'un $\forall$ positif est permis par la Ludique [Gir07], lequel vérifie de surcroît des égalités « *choquantes* » (Girard). Il convient en fait de distinguer ces deux connecteurs, qui co-existent dans $\mathsf{LK}_{\mathrm{pol}}$ et LL et qui vérifient des propriétés et des règles d'inférence distinctes : on notera $\mathbb{A}$ le second, qui, on le verra, n'est pas la « vraie » quantification.

Rappelons que le comportement de la quantification universelle est donnée plus haut par une intersection :

$$|\forall X\, A| = \bigcap_{R \in \Pi} \left(\downarrow \left| A^{\perp}\,[R/X] \right| \right)^{\perp}$$

Les remarques qui suivent s'appliquent aussi bien aux autres types issus d'une intersection, à savoir le type intersection binaire et la quantification universelle du premier ordre.

Définition de la quantification universelle : premier essai La première définition qui vient pour le comportement de la quantification universelle $\forall X\, A$ est le suivant :

$$\bigcap_{R \in \Pi} \left| A\,[R/X] \right|$$

Il s'agit bien d'un comportement par propriété de base de l'orthogonal, et cette définition correspond aux règles de déduction suivantes :

$$\frac{\vdash t : A[P/X] \mid \Gamma}{\vdash t : \exists X\, A \mid \Gamma} \qquad \frac{\vdash t : A \mid \Gamma}{\vdash t : \forall X\, A \mid \Gamma}\ (X \notin \mathcal{FV}(\Gamma))$$

Cette construction n'est cependant pas engendrée par ses valeurs en général :

Proposition 56. L'intersection de deux comportements engendrés par leurs valeurs n'est pas nécessairement engendrée par ses valeurs.

Démonstration. En voici un contre-exemple. Il suffit de prendre une observation $\perp\!\!\!\perp$ non-vide, stable par la réduction $\to_h$ et qui ne contient aucun terme mal-formé — par exemple, prenons la plus petite observation qui contient $\langle V \parallel tp\rangle$ pour tout $V \in \mathbb{V}$. On pose $T = |\mathbf{1} \oplus \top| \cap |\mathbf{0} \oplus \perp|$. Les contraintes de la proposition 48 sont remplie, donc on a $T_{\mathbb{V}} = \emptyset + \{()\}^{\perp}$. On pose alors $t = \mu\alpha.\langle \iota_2\left(\mu x.\langle \iota_1(x) \parallel \alpha\rangle\right) \Big\Vert \alpha\rangle$. De $\vdash_{LK_{pol}} t : X \oplus X^{\perp}$ on déduit par adéquation $t \Vdash \mathbf{1} \oplus \top$ et $t \Vdash \mathbf{0} \oplus \perp$. Donc $t \in T$. On pose $c_0 \in \mathcal{C}^0 \setminus \perp\!\!\!\perp$ et $u = \mu\left(\iota_1()\,.c_0 \,\middle|\, \iota_2(\alpha)\,.\langle() \parallel \alpha\rangle\right)$. On a $u \in T_{\mathbb{V}}{}^{\perp}$. Mais $\langle t \parallel u\rangle \to_h^* c_0 \notin \perp\!\!\!\perp$, donc $t \notin T_{\mathbb{V}}{}^{\perp\perp}$. □

Étant donné l'importance du théorème de génération dans la preuve du lemme d'adéquation, cela signifie qu'en dépit des apparences, un tel comportement ne s'insère pas dans les systèmes déductifs considérés, LK_{pol} et LL.

Cette remarque correspond en particulier au fait que les premières implémentations du polymorphisme en appel par valeur étaient incorrectes en présence d'effets de bords (ici, les opérateurs de contrôle de la logique classique).

Une solution : la restriction aux valeur Zeilberger revient sur ce problème et sur la façon dont on l'a résolu en introduisant une *restriction aux valeurs* [Zei08b] et montre d'une façon analogue comment retrouver de manière théorique ce phénomène au moyen d'une syntaxe focalisante.

La restriction aux valeurs consiste à modifier de façon *ad hoc* le comportement précédent afin de retrouver le théorème de génération. C'est le connecteur issu de cette restriction que l'on note $\mathbb{A}$; on pose alors :

$$|\mathbb{A}X\,A| = \left(\bigcap_{R\in\Pi} \left|A\,[R/X]\right|_{\mathbb{V}}\right)^{\perp\perp}$$

Remarquer que $\mathbb{A}X\,P$ est positif tandis que $\forall X\,P$ est négatif.

Le lemme d'adéquation s'obtient alors au prix de la restriction suivante sur les règles d'inférence :

$$\frac{\vdash t : A[P/X] \mid \Gamma}{\vdash t : \mathbb{E}X\,A \mid \Gamma} \qquad \frac{\vdash V : A \mid \Gamma}{\vdash V : \mathbb{A}X\,A \mid \Gamma}\ (X \notin \mathcal{FV}(\Gamma))$$

Ce nouveau connecteur s'insère donc dans un cadre déductif.

Shocking equalities Il vérifie cependant des propriétés hétérodoxes, par exemple que $\mathbb{A}X\,(A \oplus B)$ est le même type que $(\mathbb{A}X\,A) \oplus (\mathbb{A}X\,B)$:

Proposition 57. Soit $\perp\!\!\!\perp$ une observation non-vide, stable par la réduction $\to_\beta$ et ne contenant pas de terme mal formé. Alors les inclusions suivantes sont vérifiées :

- $|\mathbb{A}X\,(A \oplus B)| = |(\mathbb{A}X\,A) \oplus (\mathbb{A}X\,B)|$
- $|\mathbb{A}X\,(A \otimes B)| = |(\mathbb{A}X\,A) \otimes (\mathbb{A}X\,B)|$

Démonstration. Les conditions de complétude de la proposition 48 sont réunies ; on a donc :

$$\begin{aligned} & \bigcap_{R\in\Pi} \left|(A \oplus B)\,[R/X]\right|_{\mathbb{V}} \\ = & \bigcap_{R\in\Pi} \left(\left|A\,[R/X]\right|_{\mathbb{V}} + \left|B\,[R/X]\right|_{\mathbb{V}}\right) \\ = & \bigcap_{R\in\Pi} \left|A\,[R/X]\right|_{\mathbb{V}} + \bigcap_{R\in\Pi} \left|B\,[R/X]\right|_{\mathbb{V}} \end{aligned}$$

D'où le résultat pour $\oplus$. Même raisonnement pour $\otimes$. □

Ce phénomène est similaire à celui qui apparaît avec la réalisabilité intuitionniste et qui passe pour un défaut de celle-ci : celui d'admettre des principes tel $\forall X\,(A \vee B) \Rightarrow (\forall X\,A \vee \forall X\,B)$. Mais il apparaît également en Ludique : cela fait partie des « *shocking equalities* » rapportées par Girard [Gir07] ; le fait que la restriction aux valeurs ne se retrouve pas en Ludique étant compensé par la faible expressivité logique de celle-ci, qui exclut les effets de bord issus des opérateurs de contrôle de la logique classique.

Ces résultats ne sont pourtant pas surprenants lorsqu'on remarque que les preuves typiquement classiques telles que :

$$\vdash \mu\alpha.\langle \iota_1\left(\mu x.\langle \iota_2(x) \parallel \alpha\rangle\right) \Big\Vert \alpha\rangle : X \oplus X^{\perp}$$

sont de façon arbitraire exclues de cette quantification.

Comparaison des deux solutions On peut se poser la question de l'intérêt de la restriction aux valeurs en comparaison de la solution retenue pour notre définition du $\forall$: l'introduction d'un décalage. En effet, si force est de constater qu'il s'agit de deux connecteurs logiques différents, ils implémentent pourtant le même mécanisme : le décalage introduit de façon implicite avec $\forall$ sera introduit « à la main » lorsqu'il s'agira de contourner la restriction aux valeurs dans la quantification de $X \oplus X^\perp$ ci-dessus, et on dérivera $\mathbb{A}X\uparrow(X \oplus X^\perp)$ (pour l'extension triviale des types aux shifts explicites). Or on a bien pour toute formule A, $|\mathbb{A}X\uparrow A| = |\forall X\, A|$.

Il existe en fait un intérêt, du point de vue informatique, pour la restriction aux valeurs et ses équations « *choquantes* », car elles donnent davantage de règles de sous-typage. On a l'exemple des types intersection, tels qu'étudiés dans [Zei08b], pour lesquels une égalité telle que $|(P_1 \cap P_2) \otimes Q| = |(P_1 \otimes Q) \cap (P_2 \otimes Q)|$, c'est-à-dire l'égalité entre ces deux types, est souhaitée.

Du point de vue logique, la seule vraie quantification universelle est $\forall$. Transposé au système $\mathbb{F}$, cette remarque signifie que l'on considère que le λ-terme pur associé à $\Lambda X.t$ ne serait pas t mais $\lambda\_.t$, lequel est une valeur.

Cela rappelle la distinction entre *Church-style* et *Curry-style,* mais la différence entre ces connecteurs n'est pas liée au fait d'annoter les termes par des types ou non (on a été fidèle ici à la séparation entre le typage et les termes — le style de Curry).

Conclusion La distinction entre quantificateurs « *spirituels* » et « *locatifs* » proposée par Girard se retrouve jusqu'en logique classique : les « *égalités choquantes* » ne sont pas l'apanage du locatif de la Ludique, et nous obligent à nous prononcer pour la première solution. Girard propose[6] que l'on « *prenne au sérieux* » les quantificateurs « *locatifs* » et que l'on se serve de leurs « *propriétés surprenantes* ». On voit que de tels connecteurs issus de la restriction aux valeurs sont connus en informatique, où la théorie du sous-typage permet de tirer profit de ces égalités — sous-typage que la réalisabilité dévelopée ici permet d'étudier systématiquement.

Enfin, la mise en évidence du théorème de génération et, en conséquence, les considérations sur la bonne définition des comportements issus d'une intersection, permettent de lever les objections à la réalisabilité, comme le fait que celle-ci admet dans le contexte intuitionniste des principes logiques paradoxaux tel $\forall X\,(A \vee B) \Rightarrow (\forall X\, A \vee \forall X\, B)$.

[6] [Gir07], 14.3.3.

6 Conclusion

Le système L focalisant est une synthèse concise et une extension innovante de travaux distincts de la théorie de la démonstration : le $\bar{\lambda}\mu\tilde{\mu}$ de Curien-Herbelin pour l'étude de la dualité du calcul [CH00], la focalization [Plo75, Gir91], et la réalisabilité à la Krivine [Kri93, Kri04] pour le paradigme de la « programmation par preuves ». Cela fut possible grâce à la vertu unificatrice de l'explication des règles logiques de Girard à travers la Ludique et ses comportements [Gir07].

Pourtant, le résultat est étonnamment proche de la science informatique, ainsi que le montre l'analogie de l'introduction, le statut donné aux valeurs ou la présence de la distinction entre connecteurs « stricts » et connecteurs « paresseux ». Le cadre (le système L focalisant muni de la réalisabilité classique) est très impliqué et permet l'étude des langages de programmation avec effets de bord où l'ordre d'évaluation est défini.

Remerciements

Ce travail a commencé et s'est achevé au LIX et à PPS, mais c'est à Penn que la partie principale fut réalisée. Je remercie Hugo Herbelin et Pierre-Louis Curien, en raison de nombreuses discussions et commentaires, ainsi que Stephan Zdancewic ainsi que Jeffrey Vaughan pour leurs discussions.

A La logique classique constructive LC

Le système L focalisant est un quotient littéral pour LC ; voici plus de détails à ce propos. Dans ce dernier, le bénitier est interprété comme un point de retour distingué, à la façon du λ-calcul.

Groupe identité

$$\frac{}{\vdash x : P; x : P^{\perp}}\ (\alpha x+) \qquad \frac{}{\vdash \alpha : N; \alpha : N^{\perp}}\ (\alpha x-)$$

$$\frac{c : (\vdash x : N, \Gamma, \Pi)}{\vdash \mu x.c : N; \Gamma, \Pi}\ (\mu) \qquad \frac{c : (\vdash \alpha : P, \Gamma, \Pi)}{\vdash \mu\alpha.c : P \mid \Gamma, \Pi}\ (\mu)$$

$$\frac{\vdash t_- : N; \Gamma \qquad \vdash t_+ : N^{\perp} \mid \Delta, \Pi}{\langle t_+ \| t_- \rangle : (\vdash \Gamma, \Delta, \Pi)}\ (\text{N-cut}) \qquad \frac{\vdash t_+ : P; \Gamma}{\vdash t_+ : P \mid \Gamma}\ (\text{der})$$

Groupe logique

$$\frac{\vdash t : A; \Gamma \qquad \vdash u : B; \Delta}{\vdash (t, u) : A \otimes B; \Gamma, \Delta}\ (\otimes) \qquad \frac{c : (\vdash \kappa : A, \kappa' : B, \Gamma, \Pi)}{\vdash \mu(\kappa, \kappa').c : A \parr B; \Gamma, \Pi}\ (\parr)$$

$$\frac{c : (\vdash \kappa : A, \Gamma, \Pi) \qquad c' : (\vdash \kappa' : B, \Gamma, \Pi)}{\vdash \mu\left(\iota_1(\kappa).c \mid \iota_2(\kappa').c'\right) : A \,\&\, B; \Gamma, \Pi}\ (\&) \qquad \frac{\vdash t : A; \Gamma}{\vdash \iota_1(t) : A \oplus B; \Gamma}\ (\oplus_1) \qquad \frac{\vdash t : B; \Gamma}{\vdash \iota_2(t) : A \oplus B; \Gamma}\ (\oplus_2)$$

$$\frac{\vdash t : A[P/X]; \Gamma}{\vdash \{t\} : \exists X\, A; \Gamma}\ (\exists) \qquad \frac{c : (\vdash \kappa : A, \Gamma, \Pi)}{\vdash \mu\{\kappa\}.c : \forall X\, A; \Gamma, \Pi}\ (\forall)\ (X \notin \mathcal{FV}(\Gamma))$$

$$\frac{}{\vdash () : \mathbf{1};}\ (\mathbf{1}) \qquad \frac{c : (\vdash \Gamma, \Pi)}{\vdash \mu().c : \bot; \Gamma, \Pi}\ \bot$$

$$\frac{}{\vdash tp : \top; \Gamma, \Pi}\ (\top) \qquad \textit{pas de règle pour } \mathbf{0}$$

Groupe structurel Contractions, affaiblissements hors bénitier.

FIG. 9: $\mathsf{LK}_{\mathrm{pol}}$ avec bénitier explicite

Ici, on mettra en évidence le fait que le bénitier de LC est un dispositif syntaxique qui permet de définir un sous-ensemble de *valeurs*, analogue à ce que l'on fait pour les dérivés du λ-calcul en appel par valeur depuis [Plo75].

On rappelle que LC est basé sur la décomposition de $\wedge$ et $\vee$ en fonction des polarités :

$\wedge$	+	-
+	$P \otimes Q$	$N \otimes P$
-	$P \otimes N$	$N \& M$

$\vee$	+	-
+	$P \oplus Q$	$N \parr P$
-	$P \parr N$	$N \parr M$

Pour mettre en évidence le fait que L est une syntaxe pour LC, on reformule $\mathsf{LK}_{\mathrm{pol}}$ avec un bénitier, nos formules pouvant donc comporter les quatre connecteurs $\otimes, \parr, \&, \oplus$. On pourra en déduire LC en restreignant les formules à $\wedge$ et $\vee$ avec le codage ci-dessus.

$\mathsf{LK}_{\mathrm{pol}}$ avec bénitier On distingue une variable négative particulière que l'on écrit $\star$, avec la particularité qu'elle ne peut pas être sujette à l'affaiblissement ou à la contraction. On note Π un contexte vide ou de la forme $\star : P$. Il sera supposé que les contextes $\Gamma, \Delta\ldots$ ne contiennent pas $\star$. Les jugements de $\mathsf{LK}_{\mathrm{pol}}$ avec bénitier sont de la forme :

$$\begin{array}{rl} & \vdash \quad t_+ : P; \Gamma \\ & \vdash \quad t_- : N; \Gamma, \Pi \\ & \vdash \quad t_+ : P \mid \Gamma, \Pi \\ c : & (\vdash \quad \Gamma, \Pi) \end{array}$$

Le bénitier d'un séquent est soit $t_+ : P$ dans $\vdash t_+ : P; \Gamma$, soit $\star : P$ si ce dernier apparaît dans le séquent. Remarquer le point-virgule pour les termes négatifs : il s'agit d'une convention qui simplifie les règles de dérivation et qui est ana-

logue à notre choix de déclarer valeur tout terme négatif.

$\mathsf{LK}_{\mathrm{pol}}$ est formulé avec un bénitier fig. 9. Tel quel, les séquents de la forme $\vdash t : A; \Gamma, \Pi$ dérivables vérifient que t est une valeur.

Face à $\mathsf{LK}_{\mathrm{pol}}$ formulé ainsi, la première chose que l'on fait si l'on souhaite un minimum d'expressivité est de redéfinir les positifs génériques : $(t, u), \iota_i(t) \dots$ On pose alors par exemple :

$$(t, u) \stackrel{\text{def}}{=} \mu\alpha.\left\langle t \,\middle\|\, \mu x.\langle u \,\|\, \mu y.\langle (x, y) \,\|\, \alpha\rangle\right\rangle$$

qui permet la dérivation :

$$\frac{\vdash t : A \mid \Gamma \qquad \vdash u : B \mid \Delta}{\vdash (t, u) : A \otimes B \mid \Gamma, \Delta}\ (\otimes)$$

C'est là que s'opère le choix arbitraire entre gauche et droite. Cette définition justifie la règle $\rightarrow_\sigma$ pour la paire stricte, et il en va de façon similaire pour les autres constructeurs.

Termes centraux Décrire LC nécessite des règles supplémentaires pour la gestion du bénitier :

$$\frac{\vdash t_+ : P; \Gamma \qquad \vdash t_- : P^\perp; \Delta, \Pi}{\langle t_+ \,\|\, t_-\rangle : (\vdash \Gamma, \Delta, \Pi)}\ (\text{P-cut})$$

$$\frac{c : (\vdash \star : P, \Gamma)}{\vdash \mu\star.c : P; \Gamma}\ (\mu)$$

Ces règles ne permettent de dériver aucun terme supplémentaire. En revanche, elles étendent le sous-ensemble des valeurs positives en un sous-ensemble de termes *centraux* (les t_+ typables dans $\vdash t_+ : P; \Gamma$), qui se comportent comme des valeurs sans nécessairement en être. Pour de tels termes, la réduction $\langle t_+ \,\|\, \mu x.c\rangle \rightarrow' c\,[t_+/x]$ fait sens du point de vue sémantique.

Un exemple de tel terme central est $\mu\star.\langle V \,\|\, \mu\kappa.\langle V_+ \,\|\, \star\rangle\rangle$ que l'on peut lire « $\mathsf{let}\ \kappa = V\ \mathsf{in}\ V_+$ »; dans ce cas la réduction ci-dessus ne viole pas Church-Rosser. Un autre exemple de terme central, pour lequel $\rightarrow'$, viole Church-Rosser est $\mu\star.\langle \alpha \,\|\, (\star, \kappa'')\rangle$ qui compose la moitié de la dissociativité. Cela justifie que l'on rejette la réduction, quitte à la reléguer au rang de « règle η ». Enfin, remarquons que d'après [Gir91] $\mu\_.\langle V_+ \,\|\, tp\rangle$, qui se lit « $\mathcal{A}(V_+)$ », serait central.

En conclusion, cette formulation avec bénitier n'a qu'un intérêt historique (il existe d'autres manières d'identifier les termes centraux). Le bénitier est principalement un moyen peu naturel de transcrire des éléments de nature sémantique dans la syntaxe.

Références

[AH03] Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *ICALP '03*, volume 2719 of *LNCS*, 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)*, 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.

[CHM09] Pierre-Louis Curien, Hugo Herbelin, and Guillaume Munch–Maccagnoni. The duality of computation under focus. In preparation, 2009.

[DJS95] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A new deconstructive logic : Linear logic. 1995.

[Fil89] Andrzej Filinski. Declarative continuations and categorical duality. Master's thesis, 1989.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.

[Gir91] Jean-Yves Girard. A new constructive logic : Classical logic. *Math. Structures Comput. Sci*, (1), 1991.

[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.

[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, 2005.

[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.

[Plo75] Gordon D. Plotkin. Call-by-name, call-by-value and the [lambda]-calculus. *Theoretical Computer Science*, 1(2) :125–159, December 1975.

[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.

[SF93] Amr Sabry and Matthias Felleisen. Reasoning about programs in continuation-passing style. In *Lisp and Symbolic Computation*, pages 288–298, 1993.

[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.

[Zei08a] Noam Zeilberger. On the unity of duality. *Annals of Pure and Applied Logic*, 153 :1, 2008.

[Zei08b] Noam Zeilberger. Refinement types and computational duality, October 2008.