Modèles du $\lambda\mu$-calcul

Simon Castellan, Olivier Laurent, Plume, LIP

22 janvier 2014

Le contexte général

Le stage s'intègre dans l'étude des modèles des langages de programmation, en particulier le λ-calcul et le $\lambda\mu$-calcul. Ces modèles, qui représentent des invariants du calcul permettant de démontrer des résultats très abstraits sur les langages qu'il est plus difficile d'obtenir en raisonnant uniquement sur le langage. En particulier, des propriétés de cohérence logique sont souvent plus facile à montrer en exhibant un modèle de la théorie voulue. En règle générale, un modèle permet de plus d'éclairer un aspect particulier du langage.

De plus, lorsqu'on veut modéliser un langage, il faut choisir une théorie pour le langage. La tradition, notamment dans les modèles catégoriques est de considérer des théories extensionnelles, c'est-à-dire que deux termes sont égaux s'ils produisent le même résultat indépendemment du moyen employé. Au contraire, une théorie est intentionnelle si elle n'égalise que des termes qui, non seulement produisent le même résultat, mais en plus mènent les calculs de la même façon. Notre but était de mieux comprendre la question intentionnelle à la fois en λ-calcul et surtout en $\lambda\mu$-calcul.

La théorie générale des modèles du λ-calcul est très bien comprise, voir par exemple [Bar84]. Les sémantiques dénotationnelles extensionnelles du λ-calcul et du $\lambda\mu$-calcul sont très bien connues (voir [Sel01] pour le $\lambda\mu$-calcul). La question intentionnelle est, elle aussi, très connue pour le λ-calcul (voir [Bar84, Hay85]), cependant la question pour le $\lambda\mu$-calcul ne semble pas avoir été abordée.

Le problème étudié

Le problème étudié est celui des modèles du $\lambda\mu$-calcul, en particulier nous sommes intéressés par les modèles non typés et les modèles intentionnels, deux questions qui n'ont que peu d'écho dans la littérature. Les modèles typés extensionnels sont décrits par [Sel01] mais ne permettent pas complètement de parler, au contraire du λ-calcul, des théories intentionnelles du $\lambda\mu$-calcul. Il nous faut donc généraliser cette approche afin de pouvoir définir des modèles intentionnels.

À titre d'exemple de la théorie développée ici, on exprime, de façon catégorique deux modèles d'Engeler, un des premiers modèles du λ-calcul non typé intentionnel qui a de bonnes propriétés opérationnelles et on l'étend au $\lambda\mu$-calcul.

Notre cadre permet donc de mieux comprendre l'interaction entre les règles extensionnelles du $\lambda\mu$-calcul (à savoir η et θ).

La contribution proposée

Le stage apporte une nouvelle définition de modèles catégoriques pour le $\lambda\mu$-calcul (*catégories de contrôle faibles*), généralisant [Sel01] avec une preuve de correction. La séparation entre η (qui vient du

λ-calcul) et de θ (propre au $\lambda\mu$-calcul) est éclaircie et on propose un cadre dans lequel on peut définir des modèles vérifiant l'une ou l'autre indépendamment, ce qui n'est pas possible dans [Sel01]. On généralise également la méthode de [LR03] pour construire deux exemples de catégories de contrôle faibles qui peuvent se comprendre comme des modèles d'Engeler pour $\lambda\mu$. Dans cette optique, on essaye aussi d'apporter un nouveau regard sur les résultats existant en λ-calcul.

Les arguments en faveur de sa validité

La notion de modèle se base en partie sur ce qu'on sait déjà pour le λ-calcul et l'étend de manière naturelle en retrouvant les principaux théorèmes. La structure présente en λ-calcul est bien généralisée au $\lambda\mu$, et tout fonctionne comme on pourrait s'y attendre. On est capable de définir des modèles d'Engeler qui généralisent également ceux que l'on connaît pour λ. On peut également bien différencier θ et η dont la validité dépend de mécanismes indépendants (alors que l'encodage par CPS du $\lambda\mu$ dans λ écrase θ dans η).

Le bilan et les perspectives

Le stage a permis d'établir un cadre dans lequel on peut parler de modèles intentionnels en $\lambda\mu$-calcul et de séparer deux comportements extensionnels qui sont d'habitude identifiés (notamment par l'encodage par continuation), et de les étudier indépendemment l'un de l'autre. On a de plus illustré notre théorie par quelques modèles du $\lambda\mu$-calcul non typé généralisant les modèles d'Engeler pour le λ-calcul.

Il serait à présent intéressant de généraliser d'autres constructions en λ-calcul comme par exemple les types intersections avec sous-typage qui permettrait de construire des modèles vérifiant ou non η et θ de façon séparée.

Également, la notion de catégories de contrôle faibles pourrait être un peu améliorée pour être plus proche de la syntaxe, afin d'avoir un langage interne pour ces catégories.

Enfin, on pourrait également adapter ce travail à d'autres calculs proches comme $\Lambda\mu$ [Sau12].

Introduction

Le λ-calcul a été créé par Alonzo Church dans les années 1930 afin de résoudre des problèmes de calculabilité. Il a été ensuite vu comme un langage de termes pour les preuves de logique intuitionniste (c'est la correspondance de Curry-Howard), dans ce cadre les termes sont *typés*. La question des modèles à la fois dans le cadre non typé et dans le cadre typé est très importante, pour répondre à des questions de cohérence. Dans le λ-calcul typé il s'agit de cohérence logique alors que dans le λ-calcul non typé il s'agit de cohérence de théories équationnelles (une théorie équationnelle étant consistante lorsqu'il existe deux termes non égalisés par cette relation).

Au début des années 1990, M. Parigot étend le λ-calcul et la correspondance de Curry-Howard à la logique classique en introduisant le $\lambda\mu$-calcul [Par92]. La question des modèles extensionnels typés de $\lambda\mu$ est déjà bien étudiée en particulier avec les catégories de contrôle [Sel01] cependant la question des modèles non typés, elle, reste assez peu étudiée. De plus, les modèles extensionnels permettent de raisonner sur le comportement des fonctions et non pas leur implémentation. Cependant lorsqu'on étudie des langages de programmation, une théorie intentionnelle est plus adaptée car elle est souvent plus facile à décider.

On se propose donc dans ce document, d'aborder la question des modèles intentionnels et non typés du $\lambda\mu$. Dans un premier temps, nous présenterons le $\lambda\mu$-calcul et sa sémantique habituelle, dans un deuxième temps nous étudierons les modèles intentionnels pour le λ-calcul, une question dont la réponse n'est pas aussi polie que pour le cas extensionnel, et enfin nous chercherons à étendre ces notions à $\lambda\mu$.

Remerciements Je tiens à remercier Olivier pour le temps qu'il m'a consacré et toutes les discussions qu'on a pu avoir dans lesquelles j'ai appris tant. Merci à toute l'équipe Plume pour son accueil (encore une fois) chaleureux.

1 $\lambda\mu$-calcul : un langage de termes pour la déduction naturelle classique

Jusque dans les années 90, la question de l'interprétation calculatoire de la logique classique était un problème ouvert. Dans les années 90, Griffin [Gri90] se rend compte que l'on peut donner à l'opérateur `call-cc` (*Call with current continuation*) le type de la loi de Peirce : $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$. La dualité induite par la négation de la logique classique se traduit ainsi, côté programmation, par la dualité programme/continuation.

Cependant on est encore loin d'une extension de la correspondance de Curry-Howard à la logique classique. En effet, la sémantique opérationnelle de `callcc` est compliquée et ne correspond pas à une élimination de coupures. Peu après cette découverte, Michel Parigot [Par92] met au point une déduction naturelle classique et un langage de termes qui lui correspond de sorte que les coupures de la déduction naturelle soient interprétées par des réductions du langage de termes : c'est le $\lambda\mu$-calcul.

1.1 Syntaxe et typage du $\lambda\mu$-calcul

On suppose donnés deux ensembles dénombrables : $x, y, z, \ldots$ les variables de termes et α, β, γ les variables de canaux. Le λ-calcul est étendu avec la construction $\mu\alpha[\beta]t$ qui sert à interpréter la contraction et l'échange à droite de la déduction naturelle classique. La construction $\mu\alpha n$ lie α dans n. La syntaxe est la suivante :

$$\frac{}{\Gamma, x : A \vdash x : A \mid \Delta}\, ax \qquad \frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta}\, \lambda \qquad \frac{\Gamma \vdash t : A \to B \mid \Delta \qquad \Gamma \vdash u : A \mid \Delta}{\Gamma \vdash t\, u : B \mid \Delta}\, app$$

$$\frac{\Gamma \vdash t : A \mid \alpha : A, \Delta}{\Gamma \vdash [\alpha]t \mid \alpha : A, \Delta}\, raise \qquad \frac{\Gamma \vdash n \mid \alpha : A, \Delta}{\Gamma \vdash \mu\alpha\, n : A \mid \Delta}\, \mu$$

Figure 1 – Règles de typage du λμ-calcul simplement typé

$t, u ::=$		**termes**
	x	variable de terme
	$\lambda x.\, t$	abstraction de terme
	$t\, u$	application de termes
	$\mu\alpha\, n$	abstraction de canal
$n ::=$		**termes nommés**
	$[\alpha]t$	application de canal

Les types du λμ-calcul simplement typé sont les mêmes que ceux du λ-calcul simplement typé : types atomiques et types flèches. Les jugements de typage ont maintenant deux formes :

- $\Gamma \vdash t : A \mid \Delta$ où Γ est un contexte de terme (associant un type aux variables de termes) et Δ est un contexte de canal (associant un type aux variables de canaux) représentant les sorties auxiliaires de t (représentées par les sous-termes nommés de t) ;
- $\Gamma \vdash n \mid \Delta$ où Γ est un contexte de terme et Δ un contexte de canal.

Les règles sont données en figure 1. La règle *raise* interprète la contraction à droite, la règle *ax* l'affaiblissement à droite et la règle μ l'échange à droite.

1.2 Sémantique opérationnelle du λμ-calcul

Il reste maintenant à donner des règles de réduction pour ce calcul. Une intuition opérationnelle possible de ce calcul est celle des *exceptions lexicales* [Sel01]. En effet $[\alpha]t$ peut être vu comme un programme qui écrit le résultat de t sur α (d'où le fait qu'il n'ait pas de types car il ne renvoie pas de résultat) et $\mu\alpha\, n$ évalue n jusqu'à ce qu'il écrive sur le canal α et renvoie la valeur lue. Cela se distingue des exceptions que l'on trouve dans les langages de programmation habituels en cela que les noms des exceptions (variables de canaux) ne sont pas globaux et accessibles à tous mais locaux et liés à un endroit précis. Il n'est en outre pas possible d'écouter deux fois sur le même canal, à cause de la liaison.

Cette intuition donne la règle (μ) qui est la suivante :

$$(\mu) \quad (\mu\alpha\, n)u \to \mu\beta n \left[{}^{[\beta]v\, u}/_{[\alpha]v}\right] \quad (\beta \notin n, u)$$

La notation $\left[{}^{[\beta]v\, u}/_{[\alpha]v}\right]$ signifie « pour chaque sous-terme nommé de la forme $[\alpha]v$, le remplacer par $[\beta]v\, u$ ». L'idée est que lors d'une application, on ne sait pas le résultat de quel terme nommé sera lu, donc on ajoute (ou empile) u à toutes les sorties possibles sur α de n. Un μ absorbe donc tout contexte applicatif dans lequel il est placé, et en quelque sorte l'internalise (tout comme `callcc`, $\mu\alpha\, n$ capture la continuation actuelle et la stocke dans α).

On renomme α en β en faisant une étape de μ réduction pour éviter que d'éventuelles écritures sur α (libre) de u ne soient capturées après réduction. Pour que cela fonctionne, on a besoin d'une règle de renommage de canaux la ρ-règle :

$$(\rho) \quad [\beta]\mu\alpha n \to n[\beta/\alpha]$$

qui indique que lire sur α puis écrire le résultat sur β, c'est la même chose que d'écrire directement sur β quand on écrivait avant sur α. La notation $[\beta/\alpha]$ étant la substitution usuelle, qui n'opère ici que dans [_] puisque les variables de canaux ne peuvent apparaître que là. Cette règle est naturelle d'un point de vue logique car les deux preuves en déduction naturelle classique associées à ces deux termes sont les mêmes (en supposant que les contextes sont des ensembles).

Enfin, il reste une dernière règle à ajouter, une règle extensionnelle qui structure la virgule à droite, la règle θ :

$$(\theta) \quad \mu\alpha[\alpha]t = t \quad (\alpha \notin t)$$

Cette règle indique seulement que si on fait une lecture juste après une unique écriture, c'est comme ne rien faire.

Dans ce cadre, `callcc` se code ainsi par exemple

$$cc = \lambda f^{(A\to B)\to A}.\mu\alpha^A[\alpha]f(\lambda x^A.\mu\delta^B[\alpha]x)$$

Ici la continuation actuelle est capturée via le $\mu\alpha$—. cc consiste juste donc en la réification de cette continuation. On voit que les comportements classiques apparaissent lorsqu'une variable de canal possède au moins deux occurrences. (Tout $\lambda\mu$-terme clos dans lequel toute variable de canal liée apparaît exactement une fois est équivalent modulo θ à un λ-terme).

Le $\lambda\mu$-calcul *intentionnel* est le $\lambda\mu$-calcul avec seulement les règles $\beta\mu\rho$. Le $\lambda\mu$-calcul *extensionnel* est le $\lambda\mu$-calcul équipé de $\beta\mu\rho\eta\theta$ [1]. On peut également se demander quelles sont les sous-théories intéressantes entre $\beta\mu\rho$ et $\beta\mu\rho\eta\theta$. Dans la suite nous montrons comment construire des critères pour η et θ indépendants, ce qui permet de construire des modèles vérifiant l'une ou l'autre, ou aucune des deux règles extensionnelles.

***Théorème* 1.** — *Parigot Le $\lambda\mu$-calcul avec les règles $\beta\mu\rho$ est confluent et vérifie la propriété de réduction assujettie pour les règles de la figure 1. De plus, si un $\lambda\mu$-terme est simplement typé, alors il est fortement normalisant.*

1.3 Sémantique dénotationnelle extensionnelle du λμ-calcul

La question de la sémantique dénotationelle du λμ-calcul est fortement liée à celle de la logique classique. Or il est connu [GLT89] qu'un modèle dénotationnel de la logique classique est trivial : les dénotations des formules sont soit vides, soit un singleton.

Il y a plusieurs solutions pour réussir tout de même à définir une sémantique pour la logique classique, l'une d'entre elles consiste à casser la dualité conjonction/disjonction en privilégiant l'une des deux. Cette idée a donné naissance aux catégories de contrôle [Sel01] qui expliquent [2] les conséquences calculatoires

1. On pourrait considérer seulement $\beta\mu$, mais μ en l'absence de ρ se comporte très mal
2. C'est l'un des tout premiers à expliquer ce phénomène *directement* sans étudier les encodages par continuations.

de ce choix entre et/ou. Choisir la conjonction (*ie.* un produit catégorique) c'est choisir l'appel par nom, et choisir la disjonction (*ie.* une somme catégorique) c'est choisir l'appel par valeur.

Dans notre cas, la présentation du $\lambda\mu$-calcul choisie est adaptée à l'appel par nom et donc correspond aux catégories de contrôle (l'appel par valeur correspondant au dual catégorique des catégories de contrôle).

Pour interpréter le λμ-calcul simplement typé, on a besoin de trois connecteurs :

- un produit pour interpréter la virgule à gauche ;
- une flèche pour interpréter le type flèche ;
- une somme pour interpréter la virgule à droite.

Comme on travaille par appel par nom, les deux premiers points correspondent aux catégories cartésiennes closes (CCC) (voir l'annexe A.1 pour un rappel de la définition des catégories cartésienne close).

Pour interpréter la virgule à droite, on rajoute donc à la structure de CCC une opération $-\parr-$. Afin de casser la dualité produit/somme, cette opération est plus faible qu'une somme catégorique car elle n'est pas bifonctorielle, seulement fonctorielle à gauche et à droite, c'est-à-dire que si $f : A \to C$ et $g : B \to D$, $f \parr g$ n'est pas bien défini en général. Le bon cadre pour parler de telles structures, ce sont les catégories pré-monoïdales [PR97] qui arrivent naturellement lorsqu'on considère l'adjonction d'effets de bord à un langage pur (voir A.2 pour une définition et les notations associées aux catégories prémonoïdales). Par exemple, en ML, si nos fonctions peuvent faire des effets de bord, l'expression $(f\ x, g\ x)$ n'a pas de sens en elle-même, il faut spécifier lequel des deux appels est effectué en premier. Ici on a le même problème étant donné que les opérations sur les canaux peuvent être assimilées à des effets de bord.

La définition d'une catégorie de contrôle est la suivante :

***Définition* 1** (Catégorie de contrôle). — *Une* catégorie de contrôle *est une catégorie $\mathscr{C}$ qui est une CCC, munie d'une structure prémonoïdale symétrique $(\mathscr{C}, \parr, \bot)$ telle que*

- *chaque objet A ait une structure de $\parr$-monoïde (voir définition 20) choisie $(A, i_A : \bot \to A, \nabla_A : A \parr A \to A)$ de telle sorte que le choix soit cohérent avec la structure prémonoïdale. Un tel choix munit $\mathscr{C}$ d'une structure de* codiagonales *(voir déf. 21*[3]*).*
- *le par distribue sur le produit, c'est-à-dire qu'on a des isomorphismes naturels $d : (A \parr C) \times (B \parr C) \cong (A \times B) \parr C$ et $A \parr \top \cong \top$*
- *la flèche distribue sur le par, c'est-à-dire qu'on a un isomorphisme naturel $s : (B^A \parr C) \cong (B \parr C)^A$ soumis à des hypothèses de cohérence.*

Dans une catégorie de contrôle $\mathscr{C}$, on peut s'intéresser aux morphismes $f : A \to B$ « sans effet de bord », c'est-à-dire pour lesquels $f \parr g$ est toujours bien défini :

***Définition* 2** (Morphismes centraux et focaux). — *Un morphisme $f : A \to B$ d'une catégorie de contrôle $\mathscr{C}$ est dit* central *lorsque pour tout morphisme $g : C \to D$ de $\mathscr{C}$, on a $f \parr id_C; id_B \parr g = id_A \parr g; f \parr id_D$. f est dit* focal *lorsqu'il est central et qu'en outre il se comporte bien vis-à-vis de la structure de codiagonales,* ie. *les diagrammes suivants sont commutatifs :*

$$
\begin{tikzcd}
 & \bot \arrow[dl, "i_A"'] \arrow[dr, "i_B"] & \\
A \arrow[rr, "f"'] & & B
\end{tikzcd}
\qquad
\begin{tikzcd}
A \parr A \arrow[r, "f \parr f"] \arrow[d, "\nabla_A"'] & B \parr B \arrow[d, "\nabla_B"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}
$$

Un des résultats importants de [Sel01] c'est que central implique focal :

3. Qui sert à interpréter la contraction et l'affaiblissement à droite, alors qu'une structure de *diagonale* sert à interpréter la duplication et l'effacement.

Théorème 2 (Selinger). — *Tout morphisme central dans une catégorie de contrôle est focal.*

Nous avons maintenant tout ce qu'il faut pour interpréter le $\lambda\mu$-calcul simplement typé dans une catégorie de contrôle. L'interprétation est omise ici car elle sera définie plus tard dans un cadre plus général. Elle envoie un jugement $x_1 : A_1, ..., x_n : A_n \vdash t : B | \alpha_1 : C_1, ..., \alpha_m : C_m$ vers un morphisme $A_1 \times ... \times A_n \to B \parr C_1 \parr ... \parr C_m$.

Théorème 3 (Selinger). — *L'interprétation du λμ-calcul dans une catégorie de contrôle satisfait $\beta\mu\rho\eta\theta$.*

2 Modèles intentionnels et modèles non typés du λ-calcul

Les structures catégoriques sont très efficaces pour décrire des modèles extensionnels typés, mais pour décrire des modèles calculatoires sans type ou moins extensionnel il est nécessaire de travailler un peu plus.

2.1 Modèles non-typés du λ-calcul

La solution habituelle pour parler de non-typé à l'intérieur de catégories est d'utiliser des objets réflexifs.

Définition 3 (Objets réflexifs). — *Un* objet réflexif *dans une CCC $\mathscr{C}$ est un triplet (O, i, j) où O est un objet de $\mathscr{C}$ et $i : O^O \to O$, $j : O \to O^O$ sont tels que $i; j = id_{O^O}$ (ie. définissent une rétraction)*

Un tel objet réflexif est extensionnel *lorsqu'en outre $j; i = id_O$.*

À des fins de clarté, on préfère ici factoriser l'encodage usuel du λ-calcul non typé via le λ-calcul simplement typé avec type réflexif.

Définition 4 (λ-calcul simplement typé avec type réflexif). — *Le* λ-calcul simplement typé avec type réflexif *est une extension du λ-calcul simplement typé ajoutant :*

- *Un type primitif : O*
- *Deux primitives : $i : (O \to O) \to O$ et $j : O \to (O \to O)$*
- *Une règle de réduction $j(i\ t) \to t$ pour tout terme t*

On parle de λ-calcul simplement typé avec type réflexif extensionnel *lorsqu'en outre $ij(t) \to t$ pour tout terme t.*

Lemme 1. — *Tout terme du λ-calcul t dont les variables libres sont contenues dans $\{x_1, ..., x_n\}$ induit un terme t^r tel que $x_1 : O, ..., x_n : O \vdash t^r : O$ dans le λ-calcul simplement typé avec type réflexif. De plus $t =_\beta t'$ ssi $t^r =_\beta t'^r$.*

On interprète facilement le λ-calcul simplement typé avec type réflexif (resp. extensionnel) dans une CCC avec un objet réflexif (resp. extensionnel), ce qui donne par composition :

Lemme 2. — *Tout objet réflexif dans une CCC forme un modèle du λ-calcul non typé (ie. vérifie β). Si de plus l'objet considéré est extensionnel alors le modèle vérifie également η.*

En conséquence, un objet réflexif dans une CCC donne naissance à une λ-théorie, à savoir celle qui relie les termes égalisés par l'interprétation dans $(\mathscr{C}, O)$ (il est facile de voir que c'est bien une λ-théorie).

La question réciproque, à savoir est-ce que toute λ-théorie est la λ-théorie d'un objet réflexif trouve une réponse positive dans [Sco80]. La réponse est surprenante car une CCC est un objet fondamentalement extensionnel, alors comment construire une CCC à partir d'une théorie qui ne vérifie pas η ?

2.2 Catégories semi-cartésienne close

La réponse apportée par Scott est éclairée par la définition des catégories semi-cartésiennes closes donnée par Hayashi [Hay85].

***Définition* 5** (Catégorie semi-cartésienne close (version algébrique)). — *Une* catégorie semi-cartésienne close *est une catégorie avec les constructions sur les objets et morphismes suivantes :*
- *$\mathscr{C}$ a un objet $\top$, et pour chaque paire d'objets (A, B) a un objet $A \times B$ et un objet B^A,*
- *il existe deux morphismes $\pi_i : A_1 \times A_2 \to A_i$ (pour $i \in \{1, 2\}$) pour tout couple d'objets (A_1, A_2),*
- *si $x : X \to A$ et $y : X \to B$ il existe un morphisme $\langle x, y\rangle : X \to A \times B$ dans $\mathscr{C}$ tel que :*
  - *$\langle x, y\rangle; \pi_1 = x$ et $\langle x, y\rangle; \pi_2 = y$*
  - *$h; \langle x, y\rangle = \langle (h; x), (h; y)\rangle$ si $h : X' \to X$* [4]
- *il existe un morphisme $ev : B^A \times A \to B$*
- *si $f : A \times B \to C$ est un morphisme de $\mathscr{C}$ alors il existe un morphisme $\Lambda(f) : A \to C^B$ de $\mathscr{C}$ tel que*
  - *$\langle (u; \Lambda(f)), v\rangle; ev = \langle u, v\rangle; f$ pour $u : X \to A, v : X \to B$*
  - *$u; \Lambda(f) = \Lambda(\langle \pi_1; u, \pi_2\rangle; f)$ pour $u : X \to A$*
- *pour chaque objet A, on a un morphisme $\top_A : A \to \top$ tel que si $f : B \to C \in \mathscr{C}$ alors $f; \top_C = \top_B$.*

Dans une semi-CCC, les morphismes suivants sont des familles d'idempotents (ie. des morphismes $f : A \to A$ tels que $f; f = f$) qui sont tous des identités ssi $\mathscr{C}$ est une CCC :
- $\Lambda(ev) : B^A \to B^A$: correspond au λ-terme $\lambda f \lambda x.f\ x$
- $\langle \pi_1, \pi_2\rangle : A \times B \to A \times B$
- $\top_\top : \top \to \top$.

Dans un tel objet, on peut interpréter le λ-calcul simplement typé sans η. L'intérêt de cette notion c'est qu'à partir d'une semi-CCC on peut construire une CCC :

***Définition* 6** (Enveloppe de Karoubi). — *Soit $\mathscr{C}$ une catégorie. L'*enveloppe de Karoubi *(notée $\mathscr{K}(\mathscr{C})$) est la catégorie donnée par :*
- *Objets : les idempotents de $\mathscr{C}$*
- *Morphismes de $a : A \to A$ vers $b : B \to B$: les morphismes $f : A \to B$ de $\mathscr{C}$ tels que $a; f; b = f$.*

Le foncteur $J : \mathscr{C} \to \mathscr{K}(\mathscr{C})$ qui envoie A sur id_A est pleinement fidèle.

***Théorème* 4** (Hayashi). — *L'enveloppe de Karoubi d'une catégorie semi-cartésienne close est cartésienne close.*

Pour boucler la boucle, il nous faut ce résultat qui n'apparaît pourtant pas dans [Hay85] :

***Théorème* 5.** — *Soit $\mathscr{C}$ une semi-CCC et O un objet réflexif dans $\mathscr{C}$. $J(O)$ est un objet réflexif dans $\mathscr{K}(\mathscr{C})$ qui engendre la même λ-théorie que O.*

Démonstration. La rétraction de $id_O^{id_O}$ dans id_O est donnée par $(id_O^{id_O}; i; j; id_O^{id_O})$. Notons $[\![t]\!]$ l'interprétation de t dans la semi-CCC $\mathscr{C}$ et $\overline{[\![t]\!]}$ l'interprétation de t dans la CCC $\mathscr{K}(\mathscr{C})$.

L'interprétation du type τ, $\overline{[\![\tau]\!]}$ dans $\mathscr{K}(\mathscr{C})$ est par définition de l'enveloppe de Karoubi un idempotent que l'on notera $\varphi_\tau \in \mathscr{C}$. Il est facile de voir que $\varphi_\tau : [\![\tau]\!] \to [\![\tau]\!]$. On étend cette notation aux contextes en définissant un idempotent $\varphi_\Gamma : [\![\Gamma]\!] \to [\![\Gamma]\!]$ par induction sur Γ :
- $\varphi_\varepsilon = id_\top$
- $\varphi_{\Gamma,x:A} = \varphi_\Gamma \times \varphi_A$.

Dans $\mathscr{K}(\mathscr{C})$, (φ_A, φ_A) définit une rétraction de $\overline{[\![\tau]\!]}$ dans $J[\![\tau]\!]$ (idem pour les contextes). Enfin, par induction, on a que si $\Gamma \vdash t : A$ dans le λ-calcul simplement typé avec type réflexif :

4. Cette équation semble avoir été oubliée dans [Hay85].

$$\overline{[\![t]\!]} = \varphi_\Gamma; J[\![t]\!]; \varphi_A$$

Soient à présent deux termes clos t et t' du λ-calcul non typé. On a que a $\vdash t^r : O$ et $\vdash t'^r : O$ dans le λ-calcul simplement typé avec type réflexif, ce qui donne, sachant que $\varphi_\varepsilon = id_\top$ et $\varphi_O = id_O$:

$$\overline{[\![t^r]\!]} = J[\![t^r]\!] \qquad \overline{[\![t'^r]\!]} = J[\![t'^r]\!]$$

ce qui permet de conclure par fidélité de J.

□

***Corollaire* 1** (Scott). — *À toute λ-théorie correspond un objet réflexif dans une CCC qui engendre la même λ-théorie.*

Démonstration. (Ébauche)

1. Construire la catégorie syntaxique correspondant à la λ-théorie considérée, définie par :
   - Objets : décrits par la grammaire $\sigma, \tau ::= O | \sigma \to \tau$
   - Morphismes de σ vers τ : termes $t(x)$ du λ-calcul simplement typé avec type réflexif à une variable libre tels que $x : \sigma \vdash t(x) : \tau$ quotientés par l'égalité induite par la λ-théorie.
   - Composition : substitution ($t; u(x)$ est $u(t(x))$)
   - Identité : $id_\sigma(x) = x$
2. Montrer que c'est une semi-CCC avec l'encodage du produit usuel ($\sigma \times \tau = (\sigma \to \tau \to O) \to O$, $1 = O$) avec un objet réflexif, dont la λ-théorie est la même que celle de départ (lemme 1)
3. Construire l'enveloppe de Karoubi et appliquer le théorème 5

□

2.3 Modèles d'Engeler

Maintenant que nous disposons d'un cadre confortable pour discuter des modèles intentionnels du λ-calcul, il est temps d'appliquer cette théorie à des premiers exemples de modèles intentionnels : les modèles d'Engeler [Eng81]. L'intérêt de ces modèles est d'être des modèles simples intentionnels qui identifient tous les termes normalisables de tête. On en donne ici deux versions : la version habituelle (qui vit dans une catégorie de domaines) et la version relationnelle qui vit dans la catégorie des relations. Une façon d'expliquer ces modèles (et d'éviter une preuve de correction syntaxique) est de les voir comme un objet réflexif dans la catégorie de co-Kleisli d'une monade linéaire qui est cartésienne close (résultat dû à Seely [See89]).

Formulation usuelle La première formulation du modèle [Eng81] peut être vue comme un système de types à intersection.

La grammaire des types est la suivante :

$a ::= X$	atomes
$\quad \| A \to b$	arrow types
$A ::=$	**intersection types**
$\quad \Omega$	intersection vide
$\quad \| a_1 \sqcap \ldots \sqcap a_n$	

Les types sont considérés modulo associativité, commutativité et idempotence de $\sqcap$. Le type des arguments peut être vide dans un type flèche (dénoté par $\Omega \to \tau$). Il faut comprendre $a_1 \sqcap a_2 \to b$ comme le type d'une fonction qui attend un argument à la fois de type a_1 et a_2. En particulier, une fonction du type $\Omega \to b$ ne fait pas d'hypothèse sur le type de son argument.

Les jugements sont de la forme $\Gamma \vdash t : b$ où les assignements de Γ sont de la forme $x : a_1 \sqcap \ldots \sqcap a_n$. Les règles sont les suivantes :

$$\frac{}{\Gamma, x : a_1 \sqcap \ldots \sqcap a_n \vdash x : a_i} \qquad \frac{\Gamma, x : A \vdash t : b}{\Gamma \vdash \lambda x.t : A \to b} \qquad \frac{\Gamma \vdash t : A \to b \qquad \forall a \in A, \Gamma \vdash u : a}{\Gamma \vdash t\, u : b}$$

***Théorème* 6** (Engeler). — *Si $t =_\beta t'$ alors $\Gamma \vdash t : a$ est équivalent à $\Gamma \vdash t' : a$. De plus t est normalisable de tête si et seulement si il est typable par ce système.*

Cela signifie que l'ensemble des types d'un terme est invariant par β-réduction. Nous verrons dans la section suivante que ce modèle vit naturellement dans la catégorie des relations entre ensembles pré-ordonnés. La comonade ici considérée est $A \mapsto \mathscr{P}_f A$ (parties finies de A) avec la counité reliant un ensemble $X \in \mathscr{P}_f(A)$ à tout élément a tel qu'il existe $x \in X$ avec $a \leq x$.

Formulation relationnelle On peut se demander si s'intéresser aux relations entre ensembles sans ordre permet de construire un modèle similaire basé sur la catégorie des relations entre ensembles. La réponse est oui [HNPR06], mais il est un peu tordu étant donné que $\mathscr{P}_f(-)$ ne forme pas une comonade sur **Rel**. Si l'on adopte les règles suivantes :

$$\frac{}{x_1 : \Omega, \ldots, x_n : \Omega, x : a \vdash x : a} \qquad \frac{\Gamma, x : A \vdash t : b}{\Gamma \vdash \lambda x.t : A \to b} \qquad \frac{\Gamma \vdash t : a_1 \sqcap \ldots \sqcap a_n \to b \qquad \forall i, \Delta_i \vdash u : a_i}{\Gamma \cup \Delta_1 \cup \ldots \cup \Delta_n \vdash t\, u : b}$$

(où l'union de deux contextes est l'union point à point), cela ne marche pas à cause de ce défaut de naturalité, en effet, on a la dérivation suivante :

$$\frac{\vdash (\lambda y.y) : b \sqcap b \to b \quad f : a \to b, x : a \vdash fx : b \quad f : a' \to b, x : a' \vdash fx : b}{\Gamma = f : (a \to b) \sqcap (a' \to b), x : a \sqcap a' \vdash (\lambda y.y)(fx) : b}$$

(alors que $f\ x$ ne peut pas avoir le type b dans ce contexte, ce n'est donc pas un modèle pour β). Notons qu'un tel système utilisant la comonade $\mathscr{M}_f$ (multiensembles finis) sur **Rel** lui fonctionne bien, puisqu'on n'a pas $[b, b] = [b]$ dans les multiensembles. La solution est de saturer les règles :

$$\frac{}{x_1 : \Omega, \ldots, x_n : \Omega, x : a \vdash x : a} \qquad \frac{\Gamma_i, x : A \vdash t : b}{\Gamma_1 \cup \ldots \cup \Gamma_n \vdash \lambda x.t : A \to b}$$

$$\frac{\forall 1 \leq i \leq n, (\Gamma_i \vdash t : A_i \to b \qquad \forall a_i^j \in A_i, \Delta_i^j \vdash u : a_i^j)}{\bigcup_{1 \leq i \leq n} \Gamma_i \cup \bigcup_{i,j} \Delta_i^j \vdash t\, u : b}$$

Notons que ce modèle ne vérifie pas *structurellement* η car les seuls types de l'identité sont $a \to a$ alors que $\lambda f.\lambda x.f\ x$ a d'autres types comme par exemple $[(a \to b) \sqcap (a' \to b)] \to (a \sqcap a') \to b$ (type possible grâce à la saturation). Ce modèle vit donc dans une semi-CCC.

2.4 Quasi-monades et modèles intentionnels

Le but de cette section est d'expliquer le modèle d'Engeler relationnel en généralisant le résultat selon lequel la co-Kleisli d'une exponentielle au sens de la logique linéaire est cartésienne close [See89]. À la place d'une comonade linéaire, on a une quasi-comonade, dont l'exemple canonique est $\mathscr{P}_f(—) : \mathbf{Rel} \to \mathbf{Rel}$ (l'opération $x \mapsto \{x\}$ n'étant pas naturelle sur des relations non fonctionnelles).

Soit $\mathscr{C}$ une catégorie $*$-autonome avec produits finis [Mel09] (et donc coproduits finis). (On adopte la notation usuelle en logique linéaire pour la négation $—^{\perp}$, la structure monoïdale $\otimes$ et la clôture $— \multimap —$).

***Définition* 7** (Quasi-comonade). — *Une quasi-comonade sur $\mathscr{C}$ est un triplet (T, ε, δ) où $T : \mathscr{C} \to \mathscr{C}$ est un foncteur, $\delta : TA \to TTA$ est une transformation naturelle et $\varepsilon : TA \to A$ est une famille de morphismes soumis aux lois monadiques habituelles :*

– *associativité*

$$\begin{tikzcd} TTTA & TTA \arrow[l, "\delta_{TA}"'] \\ TTA \arrow[u, "T\delta_A"] & TA \arrow[l, "\delta_A"] \arrow[u, "\delta_A"'] \end{tikzcd}$$

– *identité à gauche et à droite*

$$\begin{tikzcd} TA & TTA \arrow[l, "T\varepsilon_A"'] \arrow[r, "\varepsilon_{TA}"] & TA \\ & TA \arrow[ul, "id_{TA}"] \arrow[u, "\delta_A"] \arrow[ur, "id_{TA}"'] & \end{tikzcd}$$

La sous-catégorie de $\mathscr{C}$ constituée des morphismes sur lesquels ε est naturelle est notée $\mathscr{C}_\varepsilon$.

On peut considérer la coKleisli de T, notée $Kl(T)$ définie par :

- Objets : les mêmes que ceux de $\mathscr{C}$
- Morphismes de A vers B : les morphismes de TA vers B dans $\mathscr{C}$. Un morphisme $f : TA \to B$ est noté $f : A \Rightarrow B$.
- Composition : si $f : A \Rightarrow B$ et $g : B \Rightarrow C$ leur composée, notée $f \gg g$ est donnée par $TA \xrightarrow{\delta} TTA \xrightarrow{Tf} TB \xrightarrow{g}$.

C'est une semi-catégorie car il n'y a pas d'identité (ε n'étant pas naturelle). Cependant $\varepsilon_A \in Kl(T)(A, A)$ est idempotent et donc on peut considérer $Kl(T)_\varepsilon$ la saturée d'ε, la sous-catégorie de $Kl(T)$ composée des morphismes $f : A \Rightarrow B$ tel que $\varepsilon_A \gg f \gg \varepsilon_B = f$. On a un foncteur $J : \mathscr{C}_\varepsilon \to Kl(T)_\varepsilon$ qui est l'identité sur les objets et qui à $f : A \to B \in \mathscr{C}_\varepsilon$ associe $\varepsilon_A ; f \in Kl(T)_\varepsilon(A, B)$.

***Lemme* 3** (Hyland et. al.). — *Si les projections de $\mathscr{C}$ sont dans $\mathscr{C}_\varepsilon$, alors $Kl(T)_\varepsilon$ est cartésienne et $J : \mathscr{C}_\varepsilon \to Kl(T)_\varepsilon$ préserve les produits*

***Définition* 8** (Quasi-comonade linéaire). — *T est une* quasi-comonade linéaire *lorsque c'est un foncteur monoïdal symmétrique fort de $(\mathscr{C}, \otimes, 1)$ vers $(\mathscr{C}, \times, \top)$ via $m : TA \otimes TB \cong T(A \times B) \in \mathscr{C}_\varepsilon$ et $m_1 : 1 \cong T\top \in \mathscr{C}_\varepsilon$ tel que le diagramme suivant commute :*

$$\begin{tikzcd}
T(A) \otimes T(B) \arrow[r, "m"] \arrow[dd, "\delta_A \otimes \delta_B"'] & T(A \times B) \arrow[d, "\delta"] \\
 & TT(A \times B) \arrow[d, "T\langle T\pi_1, T\pi_2 \rangle"] \\
TTA \otimes TTB \arrow[r, "m"'] & T(TA \times TB)
\end{tikzcd}$$

Le résultat suivant est suggéré mais non énoncé dans [HNPR06] :

***Théorème* 7.** — *Si les projections de $\mathscr{C}$ vivent dans $\mathscr{C}_\varepsilon$ et T une quasi-comonade linéaire, alors $Kl(T)_\varepsilon$ a une structure de semi-CCC.*

On peut appliquer ça aux catégories **Rel** (ensembles et relations entre ensembles) et $\mathbf{Rel}^{\leq}$ (ensembles pré-ordonnés et relations croissantes entre ensembles pré-ordonnés) pour obtenir respectivement une semi-CCC et une CCC qui permettent d'expliquer les deux modèles de la section précédente, avec respectivement :

- modèle relationnel : $T = \mathscr{P}_f(—)$ avec l'union pour comultiplication et $\{\{a\}, a | a \in A\}$ pour la counité $\varepsilon : \mathscr{P}_f(A) \to A$
- modèle usuel : $T = \mathscr{P}_f(—)$ (avec l'ordre usuel sur les parties $X \leq Y$ si pour tout $x \in X$ il existe $y \in Y$ tel que $x \leq y$), avec l'union pour comultiplication et $\{X, a | \exists x \in X, a \leq x\}$ pour counité.

3 Modèles intentionnels du $\lambda\mu$-calcul

3.1 Modèles non typés

Comme en λ-calcul, pour parler de modèles non typés en $\lambda\mu$-calcul on peut introduire la notion d'objets réflexifs.

***Définition* 9** (Objet réflexif dans une catégorie de contrôle). — *Un* objet réflexif dans une catégorie de contrôle $\mathscr{C}$ *est un triplet* (O, i, j) *où O est un objet de $\mathscr{C}$, $(i : O^O \to O, j : O \to O^O)$ vérifient $i; j = id_{O^O}$ et j est un morphisme focal de $\mathscr{C}$.*

Il est extensionnel quand i et j définissent un isomorphisme $O^O \cong O$.

Avec un tel objet réflexif, on peut interpréter le $\lambda\mu$-calcul non typé :

***Théorème* 8.** — *Il est possible d'interpréter le $\lambda\mu$-calcul non typé dans tout objet réflexif dans une catégorie de contrôle. L'interprétation est correcte pour $\beta\mu\rho\theta$, et correcte pour η si l'objet réflexif est extensionnel.*

La centralité de j est nécessaire pour la correction vis-à-vis de la μ-réduction. Les détails de l'interprétation seront donnés la section 3.3.

3.2 Modèles intentionnels

On souhaite à présent donner un cadre catégorique qui permet de parler de modèles intentionnels pour le $\lambda\mu$-calcul non typé. Contrairement à la situation du λ-calcul où la structure catégorique extensionnelle (CCC) permet de capturer aussi les modèles intentionnels, ce n'est plus le cas ici. Les théories engendrées par le théorème 8 contiennent toutes θ indépendamment de l'objet réflexif considéré.

Codiagonales faibles Le problème est la construction $\mu\alpha[\alpha]$— qui s'interprète comme un affaiblissement puis une contraction. L'égalité θ découle donc de la structure de monoïde de chaque type. Pour casser cette égalité, il faut que chaque type soit non plus un monoïde mais une structure plus faible. Notons que l'opération $\mu\alpha[\alpha]$— est idempotente, en effet $\mu\alpha[\alpha]\mu\alpha[\alpha]t =_\rho t$. Les types auront donc une structure plus faible que celle de monoïde mais reflétant cette idempotence :

***Définition* 10** (Monoïde faible). — *Soit $(\mathscr{C}, ⅋, \perp)$ une catégorie prémonoïdale symmétrique. Un monoïde faible dans $\mathscr{C}$ est un triplet $(A, \nabla_A : A ⅋ A \to A, i_A : \perp \to A)$ où ∇_A et i_A sont centrales et vérifient :*

– *associativité :*

$$\begin{tikzcd} A ⅋ (A ⅋ A) \arrow[r, "A⅋\nabla_A"] \arrow[d, "a"'] & A ⅋ A \arrow[dr, "\nabla_A"] & \\ (A ⅋ A) ⅋ A \arrow[r, "\nabla_A ⅋ A"'] & A ⅋ A \arrow[r, "\nabla_A"'] & A \end{tikzcd}$$

– *commutatif :*

$$\begin{tikzcd} A ⅋ A \arrow[r, "c"] \arrow[dr, "\nabla_A"'] & A ⅋ A \arrow[d, "\nabla_A"] \\ & A ⅋ A \end{tikzcd}$$

– *idempotence :*

$$\begin{tikzcd} \perp \arrow[r, "\cong"] \arrow[drr, "i_A"'] & \perp ⅋ \perp \arrow[r, "i_A ⅋ i_A"] & A ⅋ A \arrow[d, "\nabla_A"] \\ & & A \end{tikzcd}$$

Il est commutatif lorsque $c; \nabla_A = \nabla_A : A ⅋ A \to A$.

Algébriquement, un monoïde faible est un triplet $(A, \times, 1)$ où $\times$ est une opération associative et où on a $1 \times 1 = 1$. Ce qui signifie que $x \mapsto x \times 1$ est une opération idempotente :

***Définition* 11.** — *Un monoïde faible (A, ∇_A, i_A) donne naissance à un idempotent $\uparrow_A$:*

$$A \xrightarrow{\cong} A ⅋ \perp \xrightarrow{A⅋i_A} A ⅋ A \xrightarrow{\nabla_A} A$$

Cet idempotent est au cœur du problème car dans le cas d'une catégorie de contrôle, il est trivial. Cependant, θ ne parle que de la sortie principale d'un terme et pas de ses sorties auxiliaires (les canaux). L'absence de θ nous impose de relâcher la condition de monoïde sur la sortie principale, mais rien n'est dit sur les sorties auxiliaires. Si on regarde la règle de réduction de la β-réduction :

$$(\lambda x.\mu\alpha[\beta]t)\ u \to \mu\alpha[\beta]t[u/x] \quad (\alpha \notin u).$$

Cela revient à faire un affaiblissement sur la sortie α (car $\alpha \notin u$) puis une contraction à cause de l'application. Cela signifie que les sorties auxiliaires, elles, se comportent comme des monoïdes.

Dans un monoïde faible M, l'ensemble $\{x \times 1 | x \in M\}$ est le plus grand sous-monoïde faible de M qui est un monoïde. On demande donc que les sorties auxiliaires arrivent dans ce sous-monoïde :

***Lemme* 4.** — *Soit A un monoïde faible dans une catégorie prémonoïdale $\mathscr{C}$. Supposons que les deux morphismes parallèles $\uparrow_A, id_A : A \to A$ aient un égaliseur $A^\uparrow \xrightarrow{e_A} A$* [5]*. Alors $A^\uparrow$ est un monoïde faible qui est un monoïde.*

On peut donc définir la notion de *codiagonales faibles* qui généralisent les *codiagonales* de [Sel01].

***Définition* 12** (Codiagonales faibles). — *Une catégorie prémonoïdale symétrique $(\mathscr{C}, \parr, \bot)$ a des* codiagonales faibles *lorsque*

– *Chaque objet est muni d'une structure de monoïde faible commutatif choisie (A, ∇_A, i_A) compatible avec la structure prémonoïdale :*

$$\begin{tikzcd}
\bot \arrow[dr, "i_{A \parr B}"] \arrow[dd, "\cong"'] & \\
& A \parr B \\
\bot \parr \bot \arrow[ur, "i_A \parr i_B"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& A \parr B \parr A \parr B \arrow[dl, "A \parr c \parr B"'] \arrow[dd, "\nabla_{A \parr B}"] \\
A \parr A \parr B \parr B \arrow[dr, "\nabla_A \parr \nabla_B"'] & \\
& A \parr B
\end{tikzcd}$$

et $i_\bot = id_\bot$ ainsi que $\nabla_\bot : \bot \parr \bot \cong \bot$ (isomorphisme de la structure prémonoïdale)

– *Pour chaque objet A, un égaliseur de $id_A, \uparrow_A$ noté $(A^\uparrow, e_A : A^\uparrow \to A)$ est choisi de telle sorte que e_A soit focal (pour la structure de monoïde faible sur $A^\uparrow$ et A).*

Dans une catégorie avec une telle structure, on peut définir des « injections » $i_1 : A \cong A \parr \bot \xrightarrow{A \parr i_B} A \parr B$ et idem $i_2 : B \to A \parr B$ qui sont deux morphismes focaux.

Catégories de contrôle faibles Afin de terminer la définition, il nous reste à décrire l'interaction entre ces codiagonales faibles et la structure cartésienne close. Nous allons réutiliser la notion de semi-CCC bien qu'il eût été possible de définir en partant d'une CCC (nous avons vu que cela ne changeait rien aux propriétés des modèles ainsi construits) car nos exemples seront naturellement basés sur des semi-CCC. Faisant cela, nous allons devoir relaxer l'hypothèse d'isomorphisme naturel entre $(A \parr C) \times (B \parr C)$ et $A \times B \parr C$.

En effet dans une semi-CCC, l'opération $- \times -$ n'est pas un foncteur mais seulement un semi-foncteur, c'est-à-dire qu'il ne préserve que la composition (en effet $id_A \times id_B = \langle \pi_1, \pi_2 \rangle \neq id_{A \times B}$ en général), idem avec l'opération $-^-$. Cela mène à la définition suivante :

***Définition* 13** (Isomorphisme naturel entre semi-foncteurs). — *Un isomorphisme naturel entre deux semi-foncteurs $F : \mathscr{C} \to \mathscr{C}$ et $G : \mathscr{C} \to \mathscr{C}$ est une paire de transformations naturelles $(\varphi : FA \to GA, \psi : GA \to FA)$ telles que :*

– $F(id_A); \varphi_A; G(id_A) = \varphi_A$
– $G(id_A); \psi_A; F(id_A) = \psi_A$
– $\varphi_A; \psi_A = F(id_A)$ *et* $\psi_A; \varphi_A = G(id_A)$.

***Définition* 14** (Catégorie de contrôle faible). — *Une catégorie de contrôle faible est une semi-CCC $\mathscr{C}$ qui est une catégorie prémonoïdale symétrique $(\mathscr{C}, \parr, \bot)$ disposant de codiagonales faibles telles que :*

– *les projections sont focales et $i_{A \times B}; \langle \pi_1, \pi_2 \rangle = i_{A \times B}$ et $\nabla_{A \times B} = \nabla_{A \times B}; \langle \pi_1, \pi_2 \rangle$* [6]

5. Cela revient à demander à ce que l'idempotent $\uparrow_A$ soit *split*.
6. Nécessaire afin que le focus soit clos par produit.

– $\langle \pi_1 \parr C, \pi_2 \parr C\rangle : (A \times B) \parr C \to (A \parr C) \times (B \parr C)$ *fasse partie d'un isomorphisme de semi-foncteurs naturel en* A, B, C *dont l'inverse sera noté* $d : (A \parr C) \times (B \parr C) \to (A \times B) \parr C$
– $\top_{A \parr \top} : A \parr \top \to \top$ *fait partie d'un isomorphisme de semi-foncteur* $A \parr \top \cong \top$ *naturel en* A,
– *la curryfication de* $(B^A \parr C) \times A \xrightarrow{-\times i_1} (B^A \parr C) \times (A \parr C) \xrightarrow{d} (B^A \times A) \parr C \xrightarrow{ev \parr C} B \parr C$, *notée* $s : B^A \parr C \to (B \parr C)^A$ *fasse partie d'un isomorphisme de semi-foncteurs naturel en* A, B, C *dont l'inverse sera noté* s^{-1} *et soumis aux conditions de cohérence suivantes :*

$$\begin{tikzcd}
B^A \parr C^D \arrow[rr, "s'"] \arrow[d, "s"'] & & (B^A \parr C)^D \arrow[d, "s^D"] \\
(B \parr C^D)^A \arrow[r, "{s'}^A"'] & (B \parr C)^{D^A} \arrow[r, "ccc"'] & (B \parr C)^{A^D}
\end{tikzcd}$$

$$\begin{tikzcd}
B^A \parr B^A \arrow[r, "s'"] \arrow[dr, "\nabla_{B^A}"'] & (B^A \parr B)^A \arrow[r, "s^A"] & (B \parr B)^{A\times A} \arrow[dl, "\nabla_B{}^{\Delta_A}"] \\
& B^A, &
\end{tikzcd}
\qquad
\begin{tikzcd}
\bot \arrow[r, "ccc"] \arrow[d, "i_{B^A}"'] & \bot^1 \arrow[dl, "(i_B)^{1_A}"] \\
B^A. &
\end{tikzcd}$$

où $s' : B \parr C^A \xrightarrow{c} C^A \parr B \xrightarrow{s} (C \parr B)^A \xrightarrow{c^A} (B \parr C)^A$.

Lien avec les catégories de contrôle Soit $\mathscr{C}$ une catégorie de contrôle faible. On voudrait récupérer une catégorie de contrôle à partir de $\mathscr{C}$. Un problème est que pour récupérer une CCC on voudrait prendre l'enveloppe de Karoubi, hors la structure prémonoïdale de $\mathscr{C}$ ne se propage pas à $\mathscr{K}(\mathscr{C})$, parce qu'on aurait besoin de définir $a \parr b$ pour a et b deux idempotents de $\mathscr{C}$ non nécessairement centraux. Heureusement, on a :

***Lemme* 5.** — *L'ensemble des idempotents focaux est stable par produit et par exponentiation. En particulier la sous-catégorie pleine de* $\mathscr{K}(\mathscr{C})$ *restreinte aux idempotents focaux de* $\mathscr{C}$ *est cartésienne close, notée* $\mathscr{F}(\mathscr{C})$.

On a un résultat similaire pour les monoïdes de $\mathscr{C}$, c'est-à-dire les objets A tel que $\uparrow_A = id_A$:

***Lemme* 6.** — *L'ensemble des monoïdes de* $\mathscr{C}$ *est clos par produit et exponentiation.*

Il est facile de vérifier que $\mathscr{F}(\mathscr{C})$ est encore une catégorie de contrôle faible, qui est une CCC. Notons $\mathscr{C}^{\uparrow}$ la sous-catégorie pleine de $\mathscr{C}$ restreinte aux monoïdes.

***Théorème* 9.** — $\mathscr{F}(\mathscr{C}^{\uparrow})$ *est une catégorie de contrôle.*

Enfin, si A n'est pas un monoïde, alors il existe des morphismes centraux mais non focaux, comme par exemple $\theta_A : \bot^{\bot^A} \to A$ correspondant à l'élimination de la double négation.

3.3 Interprétation du $\lambda\mu$-calcul simplement typé

On peut désormais interpréter le $\lambda\mu$-calcul intentionnel dans une catégorie de contrôle faible $\mathscr{C}$. Pour cela, nous aurons besoin du morphisme suivant : $act_A : A \parr A^{\uparrow} \to A^{\uparrow}$ défini par la propriété universelle de l'égaliseur $A^{\uparrow}$ appliquée à $A \parr A^{\uparrow} \xrightarrow{A \parr e_A} A \parr A \xrightarrow{\nabla_A} A$

$$\left\llbracket \overline{\Gamma, x : A \vdash x : A|\Delta} \right\rrbracket = \llbracket\Gamma\rrbracket \times \llbracket A\rrbracket \xrightarrow{\pi_2} \llbracket A\rrbracket \xrightarrow{i_1} \llbracket A\rrbracket \parr \llbracket\Delta\rrbracket$$

$$\left\llbracket \frac{\Gamma, x : A \vdash t : B|\Delta}{\Gamma \vdash \lambda x.t : A \to B|\Delta} \right\rrbracket = \llbracket\Gamma\rrbracket \xrightarrow{\Lambda(\llbracket t\rrbracket)} (\llbracket B\rrbracket \parr \llbracket\Delta\rrbracket)^{\llbracket A\rrbracket} \xrightarrow{s^{-1}} (\llbracket B\rrbracket^{\llbracket A\rrbracket}) \parr \llbracket\Delta\rrbracket$$

$$\left\llbracket \frac{\Gamma \vdash t : A \to B|\Delta \qquad \Gamma \vdash u : A|\Delta}{\Gamma \vdash t\, u : B|\Delta} \right\rrbracket = \llbracket\Gamma\rrbracket \xrightarrow{\langle\llbracket t\rrbracket,\llbracket u\rrbracket\rangle} \llbracket(B^A \parr \Delta) \times (A \parr \Delta)\rrbracket \xrightarrow{d} \llbracket B^A \times A \parr \Delta\rrbracket \xrightarrow{ev \parr \Delta} \llbracket B \parr \Delta\rrbracket$$

$$\left\llbracket \frac{\Gamma \vdash n|\alpha : A, \Delta}{\Gamma \vdash \mu\alpha\, n : A|\Delta} \right\rrbracket = \llbracket\Gamma\rrbracket \xrightarrow{\llbracket n\rrbracket} \llbracket A\rrbracket^{\uparrow} \parr \llbracket\Delta\rrbracket \xrightarrow{e_A \parr \llbracket\Delta\rrbracket} \llbracket A\rrbracket \parr \llbracket\Delta\rrbracket$$

$$\left\llbracket \frac{\Gamma \vdash t : A|\alpha : A, \Delta}{\Gamma \vdash [\alpha]t|\alpha : A, \Delta} \right\rrbracket = \llbracket\Gamma\rrbracket \xrightarrow{\llbracket t\rrbracket} \llbracket A\rrbracket \parr \llbracket A\rrbracket^{\uparrow} \parr \llbracket\Delta\rrbracket \xrightarrow{act_A \parr \llbracket\Delta\rrbracket} \llbracket A\rrbracket^{\uparrow} \parr \llbracket\Delta\rrbracket$$

Figure 2 – Interprétation du λμ-calcul simplement typé

La définition de l'interprétation est donnée par induction sur les jugements de typage à la figure 3. Les types sont interprétés comme en λ-calcul simplement typé, les contextes à gauche également et les contextes à droite sont interprétés ainsi :

- $\llbracket\varepsilon\rrbracket = \bot$
- $\llbracket\alpha : A, \Delta\rrbracket = \llbracket A\rrbracket^{\uparrow} \parr \llbracket\Delta\rrbracket$

Un jugement de typage $\Gamma \vdash t : A|\Delta$ est interprété par un morphisme $\llbracket\Gamma\rrbracket \to \llbracket A\rrbracket \parr \llbracket\Delta\rrbracket$ de $\mathscr{C}$ et $\Gamma \vdash n|\Delta$ est interprété par un morphisme $\llbracket\Gamma\rrbracket \to \llbracket\Delta\rrbracket$.

***Théorème* 10.** — *L'interprétation donnée en figure 3 est correcte pour les règles $\beta\mu\rho$. Si $\mathscr{C}$ est une CCC, alors elle est aussi correcte pour η. Si $\mathscr{C}$ n'a que des monoïdes alors elle est aussi correcte pour θ.*

Dans ce cadre on peut aussi interpréter le $\lambda\mu$-calcul non typé :

***Définition* 15** (Objet réflexif dans une catégorie de contrôle faible). — *Un* objet réflexif dans une catégorie de contrôle faible *est un triplet $(O, i : O^O \to O, j : O \to O^O)$ avec $O \in \mathscr{C}$, $i; j = id_{O^O}$ et j focal. Il est extensionnel quand $j; i = id_O$.*

***Théorème* 11.** — *Soit (O, i, j) un objet réflexif dans une catégorie de contrôle faible. Alors on peut interpréter le $\lambda\mu$-calcul non typé dedans, et cette interprétation valide $\beta\mu\rho$. Elle vérifie en outre η si $\mathscr{C}$ est une CCC et O extensionnel et θ si O est un monoïde dans $\mathscr{C}$.*

Notons que dans ce théorème θ et η peuvent être ajoutées séparément, on peut avoir des modèles de θ ou de η seuls.

3.4 Construction de catégories de contrôle faibles

Dans cette section on cherche à étendre le résultat de la section 2.4 au $\lambda\mu$-calcul en généralisant le résultat de [LR03] qui permet de construire une catégorie de contrôle à partir d'un modèle de la logique linéaire. Pour ce faire, on considère comme à la section 2.4 une catégorie $\mathscr{C}$ qui est $*$-autonome et une quasi-comonade linéaire $T : \mathscr{C} \to \mathscr{C}$ de telle sorte que $Kl(T)_\varepsilon$ soit une semi-CCC (théorème 7) (on suppose les projections dans $\mathscr{C}_\varepsilon$).

Pour ajouter la structure de contrôle, on procède comme dans [LR03] en considérant les algèbres pour la quasi-monade duale de T, que nous noterons $T^\perp$ (définie par $T^\perp(A) = T(A^\perp)^\perp$).

On suppose également que $\mathscr{C}_\varepsilon$ est une sous-catégorie monoïdale de $\mathscr{C}$ pour les structures $(\mathscr{C}, \parr, \perp)$ et $(\mathscr{C}, \otimes, 1)$.

***Définition* 16** (Algèbre pour une quasi-monade). — *Une algèbre pour une quasi-monade (S, η, μ) sur $\mathscr{C}$ est un couple $(\underline{A}, h_A)$ où $\underline{A}$ est un objet de $\mathscr{C}$ et $h : S\underline{A} \to \underline{A}$ un morphisme de $\mathscr{C}$ satisfaisant les lois suivantes :*

$$\begin{tikzcd} \underline{A} \arrow[d, "\eta"'] \arrow[dr, "id_{\underline{A}}"] & \\ S\underline{A} \arrow[r, "h_A"'] & \underline{A} \end{tikzcd} \qquad \begin{tikzcd} SS\underline{A} \arrow[r, "\mu"] \arrow[d, "Sh_A"'] & S\underline{A} \arrow[d, "h_A"] \\ S\underline{A} \arrow[r, "h_A"'] & \underline{A} \end{tikzcd}$$

Un morphisme f de S-algèbres de A vers B est un morphisme de $\mathscr{C}$, de $\underline{A} \to \underline{B}$ tel que $Tf; h_B = h_A; f$.

Monoïdalité Les algèbres d'une quasi-monade forment une catégorie que nous noterons $T^\perp - alg$ pour le cas qui nous intéresse. Le foncteur d'oubli $\underline{\cdot} : T^\perp - alg \to \mathscr{C}$ est fidèle, ce qui permet de faire la construction suivante. On note $Kl(T)^T_\varepsilon$ la catégorie donnée par :

- Objets : les $T^\perp$-algèbres
- Morphismes : un morphisme de A vers B est un morphisme de $Kl(T)_\varepsilon$ de $\underline{A}$ vers $\underline{B}$, en d'autres termes $Kl(T)^T_\varepsilon(A, B) = Kl(T)_\varepsilon(\underline{A}, \underline{B})$.

On suppose que la monade T est monoïdale de $(\mathscr{C}, \otimes, 1)$ dans elle-même, de sorte qu'il existe un morphisme $n : TA \otimes TB \to T(A \otimes B)$ vérifiant des diagrammes de cohérence. Par dualité, cela implique que $T^\perp$ est opmonoïdale de $(\mathscr{C}, \parr, \perp)$ dans elle-même. Cela induit une structure $\parr$-monoïdale sur les algèbres de $T^\perp$:

***Lemme* 7.** — *La catégorie $T^\perp - alg$ peut être munie d'une structure monoïdale $(T^\perp - alg, \parr, \perp)$ telle que le foncteur d'oubli soit monoïdal strict (ie. $\underline{A \parr B} = \underline{A} \parr \underline{B}$).*

Démonstration. La preuve est essentiellement la même que celle de [LR03].

□

Cette structure monoïdale peut être retrouvée dans $Kl(T)^T_\varepsilon$ grâce à la promotion polarisée (comme dans [LR03])

La currification de $T(A \parr B) \otimes T(B^\perp) \xrightarrow{n} T((A \parr B) \otimes B^\perp) \xrightarrow{Tev} TA$ donne un morphisme $\varphi : T(A \parr B) \to TA \parr T^\perp B$ appelée *promotion*. La promotion polarisée est définie pour deux algèbres A et B ainsi

$$T(\underline{A} \parr \underline{B}) \xrightarrow{\varphi} T\underline{A} \parr T^\perp \underline{B} \xrightarrow{T\underline{A} \parr h_B} T\underline{A} \parr \underline{B}$$

Grâce à ce morphisme, il est facile de munir $Kl(T)^T_\varepsilon$ d'une structure prémonoïdale. De plus, si $T^\perp - alg$ a une structure de codiagonales faibles, elle se relève en une structure de codiagonales faibles sur $Kl(T)^T_\varepsilon$ (à condition que les morphismes d'affaiblissement et de contraction vivent dans $\mathscr{C}_\varepsilon$).

Semi-CCC Dans cette section, on suppose que la catégorie $T^\perp$-alg a des produits qui sont préservés par le foncteur d'oubli. Cela suffit pour importer la structure de semi-CCC de $Kl(T)_\varepsilon$:

***Lemme* 8.** — *Si $T^\perp - alg$ a des produits préservés par le foncteur d'oubli, alors $Kl(T)_\varepsilon^T$ est une semi-CCC.*

Démonstration. En effet, comme le foncteur d'oubli est fidèle, il suffit de montrer que dans $Kl(T)_\varepsilon^T$ pour deux algèbres A et B il existe une algèbre $A \Rightarrow B$ telle que $\underline{A \Rightarrow B} = \underline{A} \Rightarrow \underline{B}$ et une algèbre $A \times B$ telle que $\underline{A} \times \underline{B} = \underline{A \times B}$. Pour le premier, il suffit de prendre $T^\perp(A^\perp) ⅋ B$ car on a $\underline{T^\perp(\underline{A}^\perp) ⅋ B} = T(\underline{A}^{\perp\perp})^\perp ⅋ \underline{B} = T(\underline{A})^\perp ⅋ \underline{B}$. Pour le produit il suffit de prendre la structure de produit des algèbres qui convient car le foncteur d'oubli préserve les produits.

□

En conclusion, on a le théorème suivant :

***Théorème* 12.** — *Soit $\mathscr{C}$ une catégorie $*$-autonome avec produits finis et $T : \mathscr{C} \to \mathscr{C}$ une quasi comonade linéaire sur $\mathscr{C}$ telle que*

- *les projections de $\mathscr{C}$ vivent dans $\mathscr{C}_\varepsilon$;*
- *$\mathscr{C}_\varepsilon$ est une sous-catégorie monoïdale de $\mathscr{C}$ pour les structures $(\mathscr{C}, ⅋, \perp)$ et $(\mathscr{C}, \otimes, 1)$;*
- *T soit un foncteur monoïdal de $(\mathscr{C}, \otimes, 1)$ dans elle-même ;*
- *les algèbres de $T^\perp$ aient des produits préservés par le foncteur d'oubli, et disposent d'une structure de codiagonales faibles pour la structure monoïdale engendrée par $(\mathscr{C}, ⅋, \perp)$, telle que $\underline{\nabla_A} : \underline{A ⅋ A} \to \underline{A}$ et $\underline{i_A} : \perp \to \underline{A}$ vivent dans $\mathscr{C}_\varepsilon$.*

Alors $Kl(T)_\varepsilon^T$ est une catégorie de contrôle faible.

3.5 Modèle d'Engeler pour $\lambda\mu$

Nous pouvons à présent adapter les constructions faites en section 2.3 au $\lambda\mu$-calcul pour expliquer le modèle d'Engeler de [Lau04], en se plaçant dans **Rel** et $\mathbf{Rel}^\leq$. Dans les deux cas, il est facile de voir que les algèbres pour $\mathscr{P}_f(—)$ sont closes par produit. On munit la catégorie des algèbres de la structure de monoïde faible induite par la structure suivante :

- affaiblissement : $\perp \to \mathscr{P}_f(A)$ qui relie $*$ à n'importe quel sous-ensemble de A ;
- contraction : $\mathscr{P}_f(A) \times \mathscr{P}_f(A)$ qui relie (X, Y) à $X \cup Y$.

Cela nous donne deux catégories de contrôle faibles, une $Kl_{\mathbf{Rel}}(\mathscr{P}_f)^{\mathscr{P}_f}$ basée sur **Rel**, et $Kl_{\mathbf{Rel}^\leq}(\mathscr{P}_f)^{\mathscr{P}_f}$ basée sur $\mathbf{Rel}^\leq$. Pour avoir des modèles, il faut encore exhiber des objets réflexifs dans ces catégories.

Pour cela, il nous faut changer également la grammaire des types :

$A :=$	**types**
X	atomes
$\|I \to A$	types flèches
$U :=$	**types union**
$A_1 \sqcup \ldots \sqcup A_n (n > 0)$	
$I :=$	**types intersection**
$U_1 \sqcap \ldots \sqcap U_n$	
$\|\Omega$	

Maintenant, un terme n'a plus un type possible, mais une union de type possible, correspondant aux différentes écritures (comme les types intersections représentaient les différentes lectures de la variable).

Formellement, l'ensemble des types O vérifie $\mathscr{P}_f\mathscr{P}^*_f(O) \hookrightarrow O$ où $\mathscr{P}^*_f(O)$ sont les parties non vides de O. Il est possible de voir que $\mathscr{P}^*_f(O)$ est un objet réflexif dans $Kl(\mathscr{P}_f)^{\mathscr{P}_f}$ via $I \Rightarrow A_1 \sqcup \ldots \sqcup A_n \overset{\text{def}}{=} (I \to A_1) \sqcup \ldots \sqcup (I \to A_n)$. Cela nous donne le système de type suivant qui apparaît dans [Lau04] :

$$\frac{U \in I}{\Gamma, x : I \vdash x : U \mid \Delta} \qquad \frac{\Gamma, x : I \vdash t : U \mid \Delta}{\Gamma \vdash \lambda x.t : I \Rightarrow U \mid \Delta}$$

$$\frac{\Gamma \vdash t : (I_1 \to U_1) \sqcup \ldots \sqcup (I_n \to U_n) \mid \Delta \qquad \forall U_i \in (I_1 \sqcap \ldots \sqcap I_n), \Gamma \vdash u : U_i \mid \Delta}{\Gamma \vdash t\, u : U_1 \sqcup \ldots \sqcup U_n \mid \Delta} \qquad \frac{\Gamma \vdash t : U \mid \alpha : U', \Delta}{\Gamma \vdash [\alpha]t \mid \alpha : U \sqcup U', \Delta}$$

$$\frac{\Gamma \vdash n \mid \alpha : U, \Delta}{\Gamma \vdash \mu\alpha n : U | \Delta}$$

où $U \in I = U_1 \sqcap \ldots \sqcap U_n$ si $U = U_i$ pour un i. L'intérêt d'avoir pris les parties non vides, c'est que sinon, tout terme clos t aurait le type $\varnothing$ car $\vdash (\lambda x.x) : \varnothing \to \varnothing$ implique $\vdash (\lambda x.x)t : \varnothing$. On a comme tout à l'heure le théorème suivant [Lau04] :

Théorème **13** (Laurent). — *L'ensemble des types d'un terme est invariant par $\beta\mu\rho$ réduction. Un terme est normalisable de tête si et seulement si il est typable dans ce système.*

Notons que l'analyse catégorique faite à la section précédente permet d'obtenir la correction de ce modèle gratuitement, alors que la preuve de [Lau04] était syntaxique.

Le modèle relationnel est obtenu de la même façon, en utilisant les règles du modèle relationnel pour le λ-calcul, et en utilisant les deux règles pour $[\alpha]t$ et $\mu\alpha n$ du système précédent :

$$\frac{}{x_1 : \varnothing, \ldots, x_n : \varnothing, x : \{U\} \vdash x : U \mid \Delta} \qquad \frac{\Gamma, x : I \vdash t : U \mid \Delta}{\Gamma \vdash \lambda x.t : I \Rightarrow U \mid \Delta}$$

$$\frac{\Gamma \vdash t : (I_1 \to U_1) \sqcup \ldots \sqcup (I_n \to U_n) \mid \Delta \qquad \forall U_i \in I_1 \sqcap \ldots \sqcap I_n, \Delta_i \vdash u : U_i \mid \Delta}{\Gamma \cup \Gamma_1 \cup \ldots \cup \Gamma_n \vdash t\, u : U_1 \sqcup \ldots \sqcup U_n \mid \Delta}$$

$$\frac{\Gamma \vdash t : U | \alpha : U', \Delta}{\Gamma \vdash [\alpha]t | \alpha : U \sqcup U', \Delta} \qquad \frac{\Gamma \vdash n | \alpha : U, \Delta}{\Gamma \vdash \mu\alpha n : U | \Delta}$$

Ce système vérifie le même théorème :

Théorème **14**. — *L'ensemble des types d'un terme est invariant par $\beta\mu\rho$ réduction. Un terme est normalisable de tête si et seulement si il est typable dans ce système.*

À noter qu'en changeant la structure de codiagonales (en prenant par exemple $(x, x) \mapsto x$ comme contraction), on peut fabriquer des modèles similaires qui vérifient η et qui sont plus proches du $\lambda\mu$-calcul simplement typé.

Conclusion

En conclusion, nous avons défini une sémantique catégorique intentionnelle pour le $\lambda\mu$-calcul que nous avons utilisée pour définir des analogues des modèles d'Engeler connus pour le λ-calcul. Il reste maintenant à essayer de généraliser d'autres modèles, par exemple les modèles avec types intersection et sous-typage. Une autre piste serait d'affiner la définition de catégorie de contrôle faible pour avoir un résultat similaire à celui de Selinger sur le langage interne des catégories de contrôle. On pourrait aussi s'intéresser à $\Lambda\mu$ dont les propriétés opérationnelles sont plus intéressantes que celle de $\lambda\mu$ ($\Lambda\mu$ vérifie le théorème de séparation par exemple). Cependant le typage de $\Lambda\mu$ est moins clair que celui de $\lambda\mu$, ce qui rend l'approche catégorique délicate.

Références

[Bar84] H. P. Barendregt. *The Lambda calculus : Its syntax and semantics.* North-Holland, Amsterdam, 1984.

[Eng81] E. Engeler. Algebras and combinators. *Algebra Universalis*, 13(3) :289–371, 1981.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, April 1989.

[Gri90] Timothy G. Griffin. A formulae-as-types notion of control. In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages*, pages 47–58. ACM Press, 1990.

[Hay85] Susumu Hayashi. Adjunction of semifunctors : Categorical structures in nonextensional lambda calculus. *Theor. Comput. Sci.*, 41 :95–104, 1985.

[HNPR06] Martin Hyland, Misao Nagayama, John Power, and Giuseppe Rosolini. A category theoretic formulation for Engeler-style models of the untyped lambda calculus. *Electr. Notes Theor. Comput. Sci.*, 161 :43–57, 2006.

[Lau04] Olivier Laurent. On the denotational semantics of the untyped lambda-mu calculus. Unpublished note, January 2004.

[LR03] Olivier Laurent and Laurent Regnier. About translations of classical logic into polarized linear logic. In *LICS*, pages 11–20, 2003.

[Mel09] P.A. Mellies. Categorical semantics of linear logic. In *Interactive models of computation and program behaviour*. Société mathématique de France, 2009.

[Par92] Michel Parigot. $\lambda\mu$-calculus : an algorithmic interpretation of classical natural deduction. In *Proceedings of International Conference on Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[PR97] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7(5) :453–468, 1997.

[Sau12] Alexis Saurin. Böhm theorem and böhm trees for the λμ-calculus. *Theoretical Computer Science*, 435 :106 – 138, 2012.

[Sco80] D. S. Scott. Lambda calculus : Some models, some philosophy. In H.J. Keisler K. Kunen J. Barwise, editor, *The Kleene Symposium*, pages 223–265. North-Holland, 1980.

[See89] Robert Seely. Linear logic, $\star$-autonomous categories and cofree coalgebras. *Contemporary mathematics*, 92, 1989.

[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, April 2001.

A Catégorie de contrôle

Cette annexe définit proprement la notion de catégorie de contrôle pour fixer les notations.

A.1 Catégorie cartésienne close

***Définition* 17** (Catégorie cartésienne close). — *Une catégorie* $\mathscr{C}$ *est* cartésienne close *lorsque*

- *il existe un objet terminal* $\top$, *ie. un objet tel que pour tout objet* A *il existe un unique morphisme de* A *vers* $\top$ *(dénoté* $\top_A$*)*
- *pour tous objets* A *et* B, *il existe un produit* $(A \times B, \pi_1 : A \times B \to A, \pi_2 : A \times B \to B)$ *de* A *et* B, *ie. tel que pour tous morphismes* $f : X \to A, g : X \to B$ *il existe un unique morphisme* $\langle f, g\rangle : X \to A \times B$ *tel que* $\langle f, g\rangle; \pi_1 = f$ *et* $\langle f, g\rangle; \pi_2 = g$
- *pour tous objets* A *et* B, *il existe un exposant* $(B^A, \Lambda(—), ev : B^A \times A \to B)$ *de* A *et* B, *ie. tel que si* $f : X \times A \to B$ *alors* $\Lambda(f) : X \to B^A$ *est la* curryfiée *de* f, *et vérifie :*
  - $\Lambda(ev) = id_{B^A}$
  - $\langle \Lambda(f), g\rangle; ev = \langle id, g\rangle; f : X \to B$ *avec* $g : X \to A$.
  - $g; \Lambda(f) = \Lambda(g \times id; f)$ *pour* $g : X \to A$

A.2 Catégorie prémonoïdale

***Définition* 18** (Foncteur binoidal). — *Un foncteur binoidal* $F : \mathscr{C} \otimes \mathscr{C} \to \mathscr{C}$ *est la donnée de deux familles de foncteurs* $F_A : \mathscr{C} \to \mathscr{C}$ *et* $F^B : \mathscr{C} \to \mathscr{C}$ *où* A, B *sont des objets de* $\mathscr{C}$, *tels que* $F_A(B) = F^B(A) \overset{\text{def}}{=} F(A, B)$. *Si* $f : A \to B$, *on pose* $F(f, C) = F_C(f) : F(A, C) \to F(B, C)$ *et* $F(C, f) = F^C(f) : F(C, A) \to F(C, B)$.

Un morphisme $f : A \to B$ est *central* pour un foncteur binoidal lorsque pour tout $g : C \to D$, $F(f, C); F(B, g) = F(A, g); F(f, D)$ et $F(C, f); F(g, B) = F(g, A); F(D, f)$.

***Définition* 19** (Catégorie prémonoïdale). — *Une structure prémonoïdale sur une catégorie* $\mathscr{C}$ *est donnée par*

- *un foncteur binoidal* $— \parr — : \mathscr{C} \otimes \mathscr{C} \to \mathscr{C}$
- *un objet* $\perp$ *de* $\mathscr{C}$
- *deux isomorphismes naturels centraux en* A, $l : A \cong A \parr \perp$ *et* $r : A \cong \perp \parr A$
- *un isomorphisme naturel en* A, B, C, $a : (A \parr B) \parr C \cong A \parr (B \parr C)$

soumis aux diagrammes de cohérence usuels pour les catégories monoïdales [PR97].

Elle est symétrique quand de plus, on a un isomorphisme naturel $c : A \parr B \cong B \parr A$ *cohérent vis-à-vis de* l, r *et* a.

***Définition* 20** (Monoïde). — *Un* $\parr$*-monoïde dans une catégorie prémonoïdale* $(\mathscr{C}, \parr, \perp)$ *est un triplet* $(A, i_A : \perp \to A, \nabla_A : A \parr A \to A)$ *où* i_A *et* ∇_A *sont deux morphismes centraux tels que*

- *associativité :*

$$\begin{tikzcd}
A \parr (A \parr A) \arrow[r, "A \parr \nabla"] \arrow[d, "a"'] & A \parr A \arrow[dr, "\nabla"] & \\
(A \parr A) \parr A \arrow[r, "\nabla \parr A"'] & A \parr A \arrow[r, "\nabla"'] & A
\end{tikzcd}$$

- *identité :*

$$\begin{tikzcd}
A \parr \perp \arrow[r, "A \parr i_A"] \arrow[dr, "l^{-1}"'] & A \parr A \arrow[d, "\nabla_A"'] & \perp \parr A \arrow[l, "i_A \parr A"'] \arrow[dl, "r^{-1}"] \\
& A &
\end{tikzcd}$$

Il est commutatif lorsque

$$\begin{tikzcd}
A \parr A \arrow[r, "c"] \arrow[dr, "\nabla"'] & A \parr A \arrow[d, "\nabla"] \\
& A \parr A
\end{tikzcd}$$

commute.

Définition **21** (Codiagonales). — *Une catégorie prémonoïdale symmétrique* $(\mathscr{C}, \parr, \perp)$ *a des codiagonales lorsque chaque objet est muni d'une structure de* $\parr$*-monoïde commutatif* (A, i_A, ∇_A) *tel que :*

- $i_\perp = id_\perp : \perp \to \perp$
- *on ait :*

$$\begin{tikzcd}
\perp \arrow[dr, "i_{A \parr B}"] \arrow[dd, "l=r"'] & \\
& A \parr B \\
\perp \parr \perp \arrow[ur, "i_A \parr i_B"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& A \parr B \parr A \parr B \arrow[dl, "A \parr c \parr B"'] \arrow[dd, "\nabla_{A \parr B}"] \\
A \parr A \parr B \parr B \arrow[dr, "\nabla_A \parr \nabla_B"'] & \\
& A \parr B
\end{tikzcd}$$

A.3 Catégorie de contrôle

Définition **22** (Catégorie de contrôle). — *Une catégorie de contrôle est une catégorie prémonoïdale* $(\mathscr{C}, \parr, \perp)$ *avec codiagonales qui est une CCC, avec de plus*

- $\langle \pi_1 \parr C, \pi_2 \parr C \rangle : A \times B \parr C \to (A \parr C) \times (B \parr C)$ *est un isomorphisme naturel en* A, B, C.
- *la curryfication de* $(B^A \parr C) \times A \xrightarrow{-\times(l; A \parr i_C)} (B^A \parr C) \times (A \parr C) \xrightarrow{d} B^A \times A \parr C \xrightarrow{ev \parr C} B \parr C$ *définit un morphisme* $s : (B^A \parr C) \to (B \parr C)^A$ *qui est un isomorphisme naturel en* A, B, C *vérifiant les deux diagrammes suivant :*

$$\begin{tikzcd}
B^A \parr C^D \arrow[rr, "s'"] \arrow[d, "s"'] & & (B^A \parr C)^D \arrow[d, "s^D"] \\
(B \parr C^D)^A \arrow[r, "s'^A"'] & (B \parr C)^{D^A} \arrow[r, "ccc"'] & (B \parr C)^{A^D}
\end{tikzcd}$$

$$\begin{tikzcd}
B^A \parr B^A \arrow[r, "s'"] \arrow[dr, "\nabla_{B^A}"'] & (B^A \parr B)^A \arrow[r, "s^A"] & (B \parr B)^{A \times A} \arrow[dl, "\nabla_B^{\Delta_A}"] \\
& B^A, &
\end{tikzcd}
\qquad
\begin{tikzcd}
\perp \arrow[r, "ccc"] \arrow[d, "i_{B^A}"'] & \perp^1 \arrow[dl, "(i_B)^{1_A}"] \\
B^A. &
\end{tikzcd}$$

où $s' : B \parr C^A \xrightarrow{c} C^A \parr B \xrightarrow{s} (C \parr B)^A \xrightarrow{c^A} (B \parr C)^A$.