

Vers une sémantique complètement adéquate de l'appel-par-valeur, à l'aide de la logique linéaire différentielle

Giulio Manzonetto

1 Contexte

1.1 Le λ -calcul et la logique linéaire différentielle

Le λ -calcul est un système formel qui joue un rôle central en logique et en informatique théorique puisque, grâce à sa syntaxe simple et puissante, il représente le contexte idéal pour étudier la notion même de “calcul”. Le λ -calcul constitue le noyau de tout langage de programmation fonctionnel, il est donc naturel de lui associer des stratégies de réduction spécifiant l'ordre dans lequel les différentes sous-parties d'un programme, appelé dans ce contexte λ -terme, s'évaluent. Le λ -calcul “classique” peut être considéré en appel-par-nom : le sous-terme le plus à gauche est évalué d'abord. Dans ce cadre, la notion de *continuité syntaxique* a été la clef qui a permis le développement d'une théorie de l'approximation des programmes permettant d'étudier le comportement infini d'un λ -terme, représenté par son arbre de Böhm, en regardant ses portions finies [Bar84, §10]. Cette intuition sur la notion de calcul s'est répercutée aussi sur les types de modèles dénotationnels introduits, qui modélisent une donnée comme un élément d'un domaine (cpo) et un programme comme une fonction (Scott-)continue entre deux domaines. Le modèle \mathcal{D}_∞ de Scott a aussi la propriété d'être complètement adéquat : deux λ -termes ont la même interprétation si et seulement si ils sont observationnellement équivalents.

Un changement important de point de vue a été apporté grâce à l'introduction de la logique linéaire [Gir87], et plus récemment de son extension différentielle [ER06]. L'outil principal de cette dernière est un opérateur de dérivation syntaxique qui permet de transformer l'application du λ -calcul en une somme formelle d'applications linéaires dans lesquelles l'argument remplace seulement une occurrence de la variable de la fonction. Ces sommes formelles ont une correspondance sémantique et une interprétation mathématique tout à fait canonique : elles expriment dans la syntaxe l'addition vectorielle des sémantiques quantitatives de la logique linéaire et elles correspondent aux sommes que l'on obtient dans l'analyse lorsqu'on calcule la dérivée d'un produit de fonctions. Le travail d'Ehrhard et Regnier a donc permis de passer d'une étude qualitative des propriétés extensionnelles de programmes (comme la terminaison) à une étude quantitative (terminaison en un certain nombre d'étapes, quantité de ressources utilisées pendant l'évaluation) en remplaçant les arbres de Böhm par la notion de développement de Taylor

des λ -termes. La série formelle qu'on obtient par ce développement a une interprétation calculatoire très intéressante : elle permet d'analyser la normalisation d'un λ -terme par le biais de la normalisation de termes d'un λ -calcul avec ressources, une version affaiblie du λ -calcul où l'utilisation d'une ressource est linéaire et le nombre de ressources qu'un terme peut utiliser est borné. Ce langage peut être interprété naturellement dans la sémantique relationnelle, qui constitue le modèle quantitatif le plus simple de la logique linéaire. Des résultats importants font le lien entre l'approche qualitative et quantitative :

deux λ -termes ont le même arbre de Böhm
 \iff
les formes normales de leur développements de Taylor coïncident
 \iff
ils ont la même interprétation dans un certain modèle relationnel.

1.2 Le λ -calcul en appel-par-valeur

La stratégie d'évaluation appel-par-nom présente l'inconvénient d'être généralement très inefficace. Pour cette raison, la plupart des langages de programmation utilise l'appel-par-valeur : on évalue les arguments d'abord et ensuite le corps de la fonction. Le λ -calcul en appel-par-valeur est connu depuis 1975 [Plo75] mais sa théorie est beaucoup moins développée par rapport à celle du λ -calcul appel-par-nom. Par exemple, aucun modèle dénotationnel connu n'est complètement adéquat : il existe des modèles adéquats satisfaisant des théorèmes d'approximation (voir [RDRP04]), mais les systèmes d'approximations utilisés ne sont pas satisfaisant selon les auteurs eux-mêmes. Le problème de définir une bonne notion d'approximation se répercute aussi du côté syntaxique : aucune notion d'arbre de Böhm n'avait été proposée en littérature à cause de certaines réductions qui restent bloquées en attendant des valeurs. Seulement récemment cette théorie a fait des progrès remarquables : d'un côté, Carraro et Guerrieri [CG14] ont introduit une règle de calcul inspirée par les réseaux de preuves qui permet de débloquent plusieurs de ces redex ; de l'autre côté Ehrhard a montré que la notion de développement de Taylor passe au cadre en appel-par-valeur sans problèmes particuliers [Ehr12]. Pendant son stage de Master 2, sous la direction de Manzonetto et Pagani, Kerinec a exploité ces outils afin de donner une définition d'arbre de Böhm adaptée à ce type de stratégie d'évaluation. Le critère décisif qui permet d'affirmer que cette notion est satisfaisante est le fait qu'on retrouve le théorème d'Ehrhard et Regnier reliant arbres de Böhm et développement de Taylor : la forme normale du développement de Taylor d'un λ -terme coïncide avec le développement de Taylor de son arbre de Böhm.

2 Objectifs

Le travail effectué par Kerinec pendant son stage de Master 2, qui a fait l'objet d'un article actuellement soumis [KMP18], représente un point de départ solide pour une thèse de doctorat. Le but de la thèse est de renforcer et généraliser la théorie de l'approximation de programmes de plusieurs manières dans le cadre de l'appel-par-valeur.

2.1 Sémantique relationnelle et variations

Le point de départ choisi pour faire une analyse sémantiques est le modèle relationnel de la logique linéaire différentielle. En effet, ce modèle constitue la sémantique quantitative la plus simple du λ -calcul en appel-par-valeur, qui est néanmoins très intéressante puisque tout élément de l'interprétation d'un terme M contient des informations sur le nombre de ressources utilisées par M pendant sa réduction ainsi que sur le nombre d'étapes nécessaires pour produire une valeur. Dans l'article [KMP18], a été conjecturé que deux λ -termes ayant le même arbre de Böhm en appel-par-valeur partagent aussi la même interprétation dans le modèle relationnel introduit par Ehrhard [Ehr12] et *vice versa*. L'intérêt d'arriver à démontrer ce résultat est de pouvoir étudier cette équivalence entre programmes indifféremment avec des techniques syntaxiques (développement de Taylor et arbres de Böhm) ou bien sémantiques (modèles relationnels).

Dans l'article [MPR19], Manzonetto, Pagani et Ronchi Della Rocca définissent des modèles du λ -calcul en appel-par-valeur à l'aide d'un système de types construits à partir d'un ensemble d'atomes en utilisant la flèche \multimap et le tenseur symétrique \otimes de la logique linéaire. D'habitude, ces genres de systèmes de types correspondent à des modèles relationnels mais dans ce cas la situation n'est pas si claire : les règles de typage ressemblent plutôt à celles d'un système en appel-par-nom, mais un élément idempotent est injecté dans l'interprétation de toute valeur. Dans l'esprit, cette construction utilise une monade de lifting (sauf que la catégorie des ensembles et de relations n'admet pas de lifting) et partage des similitudes avec les relations pointées de [BDER97]. Il est certainement intéressant d'éclaircir le cadre catégorique sous-jacent à la construction de ces modèles pour pouvoir appliquer des théorèmes puissants de théorie des catégories afin d'étudier leurs propriétés abstraites.

2.2 Vers un modèle complètement adéquat

Dans l'article [MPR19], les auteurs démontrent que tous ces modèles sont adéquats pour le λ -calcul en appel-par-valeur et satisfont un théorème d'approximation pour le système d'approximations qui est à la base des arbres de Böhm introduits dans [KMP18]. En analysant l'élimination des coupures de ces systèmes de types, il est évident que ces modèles valident une règle de réduction plus permissive que celle standard : il est possible de contracter un redex $(\lambda x.M)N$ en préservant l'interprétation du programme, même lorsque N n'est pas une valeur, à condition que la variable x apparaisse effectivement dans M et que ses occurrences satisfassent certaines propriétés. De ce fait, les arbres de Böhm ne constituent pas un modèle (syntaxique) complètement adéquat du λ -calcul en appel-par-valeur : il existe des λ -termes observationnellement équivalents qui ont des arbres de Böhm différents. D'un côté, la situation est spéculaire au cas en appel-par-nom, où il faut introduire des arbres de Nakajima pour avoir un modèle complètement adéquat [Cur02]. De l'autre côté, cela montre qu'il n'est pas suffisant de considérer les arbres de Böhm modulo une équivalence extensionnelle afin de capturer l'équivalence observationnelle entre programmes. On souhaite analyser quels types de réductions sont légales afin de raffiner la notion d'arbres de Böhm et induire toutes les égalités validées

par ces modèles, seulement ensuite on pourra rajouter des η -expansions appropriées et arriver à une notion satisfaisante d'arbres de Nakajima en appel-par-valeur. De cette façon, on compte pouvoir définir le premier modèle complètement adéquat de ce calcul.

2.3 Extensions non-déterministes

Une fois que la situation a été éclaircie dans le cas classique, nous souhaitons analyser des extensions non-déterministes du λ -calcul [DdP93]. Il est en fait possible d'enrichir la syntaxe du λ -calcul en rajoutant des opérateurs non-déterministes comme un choix interne et un opérateur parallèle. Cela permet d'écrire des programmes qui peuvent calculer des valeurs différentes, selon les choix effectués pendant la réduction, où bien calculer plusieurs résultats en parallèle. Dans la sémantique relationnelle les λ -termes sont dénotés par des relations et cela permet d'interpréter le choix non-déterministe simplement par l'union ensembliste. Certains modèles satisfont aussi la règle *mix* de la logique linéaire, qui permet d'interpréter de façon naturelle la composition parallèle comme fait dans [BEM12] en appel-par-nom et dans [DMP13] en appel-par-valeur. Notre objectif est de se faire guider par cette sémantique afin de développer des notions d'arbres de Böhm non-déterministes, pour ces deux stratégies d'évaluation.

Références

- [Bar84] H. P. Barendregt. *The lambda-calculus, its syntax and semantics*. Number 103 in *Studies in Logic and the Foundations of Mathematics*. Second edition, 1984.
- [BDER97] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Believe it or not, ajm's games model is a model of classical linear logic. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 68–75. IEEE Computer Society, 1997.
- [BEM12] A. Bucciarelli, T. Ehrhard, and G. Manzonetto. A relational semantics for parallelism and non-determinism in a functional setting. *Ann. Pure Appl. Logic*, 163(7) :918–934, 2012.
- [CG14] A. Carraro and G. Guerrieri. A semantical and operational account of call-by-value solvability. In A. Muscholl, editor, *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of ETAPS, volume 8412 of Lecture Notes in Computer Science*, pages 103–118. Springer, 2014.
- [Cur02] P.-L. Curien. Sur l'eta-expansion infinie, 2002. *Comptes Rendus de l'Académie des Sciences* 334, Sec. I, 77-82.
- [DdP93] M. Dezani-Ciancaglini, U. de'Liguoro, and A. Piperno. Filter models for a parallel and non deterministic lambda-calculus. In A.M. Borzyszkowski and S. Sokolowski, editors, *Mathematical Foundations of Computer Science 1993, 18th International Symposium, MFCS'93*, volume 711 of *Lecture Notes in Computer Science*, pages 403–412. Springer, 1993.

- [DMP13] A. Díaz-Caro, G. Manzonetto, and M. Pagani. Call-by-value non-determinism in a linear logic type discipline. In S. N. Artëmov and A. Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 164–178. Springer, 2013.
- [Ehr12] T. Ehrhard. Collapsing non-idempotent intersection types. In P. Cégielski and A. Durand, editors, *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 259–273, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [ER06] T. Ehrhard and L. Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2) :166–195, November 2006. Special issue on Logic, Language, Information and Computation, 11th Workshop on Logic, Language, Information and Computation.
- [Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [KMP18] E. Kerinec, G. Manzonetto, and M. Pagani. Revisiting call-by-value Böhm trees in light of their Taylor expansion. *CoRR*, abs/1809.02659, 2018. Soumis.
- [MPR19] G. Manzonetto, M. Pagani, and S. Ronchi Della Rocca. Some relational models for call-by-value λ -calculus, 2019. Accepted in *Fundamenta Informaticae*.
- [Plo75] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theor. Comput. Sci.*, 1(2) :125–159, 1975.
- [RDRP04] S. Ronchi Della Rocca and L. Paolini. *The Parametric λ -Calculus : a Meta-model for Computation*. EATCS Series. Berlin, 2004.

Encadrement

Giulio Manzonetto (Directeur), giulio.manzonetto@lipn.univ-paris13.fr

Laboratoire d'accueil : LIPN - Laboratoire d'Informatique de Paris Nord (UMR 7030 du CNRS). Université Paris 13, Villetaneuse.

Le Laboratoire d'Informatique de Paris Nord (LIPN - UMR 7030) joue un rôle majeur dans la recherche en informatique fondamentale en Île-de-France. En particulier, le laboratoire accueille, dans l'équipe LoVe (Logique et Vérification), un groupe de chercheurs et enseignant-chercheurs ayant une expertise bien établie et reconnue de longue date en logique linéaire, avec des applications à plusieurs domaines de l'informatique fondamentale : théorie de la démonstration, lambda-calcul et programmation fonctionnelle, sémantique dénotationnelle, complexité implicite. Ces domaines font partie des axes portants de l'équipe LoVe et du LIPN.

À présent, 2 professeurs, 5 maîtres de conférences et 3 CR CNRS travaillent activement sur ces thématiques, et le groupe conduit ou contribue à plusieurs projets de

recherche nationaux autour du sujet, et mène régulièrement des collaborations internationales avec, entre autres, l'Italie, la Grande-Bretagne, le Japon, les Pays-Bas et la Danemark.