Paramétricité pour les langages avec effets

4 février 2019

Il s'agit d'un sujet pour les étudiants avec un goût prononcé pour les mathématiques, qui ont aimé le cours de PROG et/ou dont le séminaire de théorie des catégories ou la correspondance de Curry-Howard ont suscité la curiosité.

Un joli post de blog, http://prl.ccs.neu.edu/blog/2017/06/05/syntactic-parametricity-strikes-again/, attirait notre attention sur le fait, semble-t-il peu connu, que les techniques de recherche de preuve permettent d'obtenir très facilement des théorèmes de *paramétricité*. C'est lié à une technique simple et élégante expliquée dans Girard et al. (1992) qui mériterait effectivement d'être mieux connue et expliquée aux gens qui s'intéressent à la théorie des langages de programmation (qui bien souvent on des *effets de bord* : non-terminaison, opérateurs de contrôle, état, exceptions...).

La paramétricité est à la base de l'interprétation calculatoire du système F (le λ-calcul polymorphe). Par exemple, pour tous termes clos t, u du système F de types respectifs $\forall\alpha((A \to \alpha) \to \alpha)$ et $A \to B$, l'analyse des formes normales donne : $u(t(\lambda y.y)) = tu$. Cela implique, dans les modèles intéressants de système F, l'isomorphisme $\forall\alpha((A \to \alpha) \to \alpha) \cong A$, à la base de l'interprétation des type de données dans le système F, par exemple :

$$\forall\alpha((A \to \alpha) \to (B \to \alpha) \to \alpha) \cong \forall\alpha(((A + B) \to \alpha) \to \alpha) \cong A + B$$

autrement dit, le type des booléens de Church est bien un type booléen ! et autres propriétés du genre.

L'idée de Girard et al. (1992) est très simple : on montre que la propriété est vraie pour les formes normales, puis on montre que les termes normalisent. On montre en fait un résultat de paramétricité pour tous les jugements de types simples $F(\alpha) \vdash G(\alpha)$ à la fois, en énonçant avec le langage de la théorie des catégories que ce sont des transformations *dinaturelles* (ci-dessus, exemple avec $F(\alpha) = A \to \alpha$ et $G(\alpha) = \alpha$). Les transformations dinaturelles forment presque un modèle, il leur manque juste la composition ! C'est là que la normalisation (= élimination de la composition) intervient. Cela établit en réalité un lien fondamental entre 1) paramétricité dans les langages de programmation, 2) élimination des coupures en théorie de la démonstration, et 3) théorèmes de cohérence en théorie des catégories.

Toute la partie de théorie de la démonstration peut être reformulée de façon moderne, auto-suffisante, concise et rigoureuse grâce à des techniques récentes (Munch-Maccagnoni, 2017, Thms 55, 56, 66, 67). Mais il y a mieux ! En faisant cela, on se rend compte qu'on n'utilise pas toutes les équations du langage d'origine. Le travail pré-cité caractérise les équations dont on a besoin, en les rapprochant des modèles du λ-calcul avec effets de bord, qui forcent un ordre d'évaluation (un modèle standard appelé le call-by-push-value, *CBPV*). Par conséquent, cela mène a généraliser le résultat de paramétricité de Girard et al. (1992) aux langages avec effets. Toute la difficulté est de savoir comment étendre l'énoncé du résultat dans ce nouveau cadre. Cela devrait se passer comme dans Levy (2017) qu'on pourra retrouver comme un cas particulier. On peut également s'en inspirer pour réfléchir à l'implication du résultat pour des cas d'effets concrets (non-terminaison, etc.).

Le stage est en deux étapes. La première étape est bibliographique et consiste à comprendre ce que j'ai raconté ci-dessus. Le résultat sera une reformulation concise et avec des techniques modernes du résultat de Girard et al. (1992). La deuxième consiste en une généralisation aux modèles avec effets de bord. Le but est d'obtenir un résultat qui généralise à la fois Girard et al. (1992) et Levy (2017). Les connaissances acquises seront utiles à n'importe qui souhaite poursuivre dans la théorie des langages de programmation, la théorie de la démonstration, ou la théorie des catégories.

Références

Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. 1992. Normal Forms and Cut-Free Proofs as Natural Transformations. In *in : Logic From Computer Science, Mathematical Science Research Institute Publications 21*. Springer-Verlag, 217–241.

Paul Blain Levy. 2017. Contextual isomorphisms. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. ACM, 400–414.

Guillaume Munch-Maccagnoni. 2017. *Note on Curry's style for Linear Call-by-Push-Value*. Technical Report. https://hal.inria.fr/hal-01528857