Une sémantique catégorique de la logique de séparation concurrente

Stage sous la direction de Paul-André Melliès (PPS, IRIF)

La logique de séparation [9] est née à la fin des années 1990 de la description sémantique par O'Hearn et Pym [8] de certaines catégories de préfaisceaux sur une catégorie monoïdale, dont les objets décrivent un espace mémoire disponible. Ces catégories de préfaisceaux ont la propriété particulière de disposer d'un produit tensoriel défini par convolution (appelé produit de Day) ainsi que d'un produit cartésien qui correspond à la conjonction de la logique usuelle. La logique de séparation a découlé de ces travaux de nature algébrique. Il s'agit d'une logique de Hoare dont les formules sont construites au moyen des combinateurs traditionnels de la logique classique, ainsi que d'un produit tensoriel (la conjonction séparante notée $*$) qui permet de raisonner de manière modulaire sur l'espace mémoire utilisé par des programmes.

On dispose aujourd'hui de nombreuses variantes de logique de séparation, en particulier de plusieurs extensions de cette logique de Hoare dédiée à la preuve de programmes concurrents. La validité de ces logiques repose sur un théorème de correction (soundness theorem) qui assure que toute spécification établie dans la logique pour un programme donné sera bien satisfaite lors de l'exécution. Il est connu que ces théorèmes de correction sont particulièrement difficiles à établir dans le cas des logiques de séparation concurrentes.

Le but de ce stage sera de formuler une sémantique catégorique de la logique de séparation concurrente, et d'en déduire un théorème de correction pour cette logique. On s'appuiera pour cela sur les travaux récents de Melliès et Zeilberger [7] sur les systèmes de raffinement de type. Ces travaux offrent en effet un pont entre théorie de la démonstration et logique de spécification, telle que la logique de Hoare ou la logique de séparation. Le cadre offert par les systèmes de raffinement de type permet en particulier de définir le produit tensoriel de Day sur une catégorie monoïdale, et de reconstruire ainsi les briques de base de la logique de séparation traditionnelle. L'une des ambitions du stage sera donc d'étendre cette correspondance à la logique de séparation concurrente tout entière.

Nous partirons pour cela des articles [6, 10] qui ont corrigé la première démonstration (fausse) de correction de la logique de séparation concurrente,

donnée par Brookes [4, 5] en 2004 puis 2007. Si le temps le permet, nous étudierons dans ce cadre le cas des logiques de séparation d'ordre supérieure [1, 2, 3].

References

[1] Bodil Biering, Lars Birkedal, and Noah Torp-Smith. BI-Hyperdoctrines, Higher-order Separation Logic, and Abstraction. ACM Trans. Program. Lang. Syst., 5:29, 2007.

[2] Lars Birkedal, Noah Torp-Smith, and Hongseok Yang. Semantics of Separation-Logic Typing and Higher-order Frame Rules for Algol-like languages. LMCS, 5:2, 2006.

[3] Lars Birkedal, Bernhard Reus, Jan Schwinghammer, and Hongseok Yang. Simple Model of Separation Logic for Higher-order Store. In Proceedings of ICALP 2008.

[4] Stephen Brookes. A semantics for concurrent separation logic. Proceedings of CONCUR 2004.

[5] Stephen Brookes. A semantics for concurrent separation logic. Festschrift for John Reynold's 70th Birthday, TCS Vol. 375, May 2007.

[6] Stephen Brookes. A revisionist history of concurrent separation logic. In Proceedings MFPS, 2011.

[7] Paul-André Melliès, Noam Zeilberger. Functors are type refinement systems, ACM Symposium on Principles of Programming Languages, POPL 2015.

[8] P.W. O'Hearn and D. J. Pym. The logic of bunched implications Bulletin of Symbolic Logic , 5(2), June 1999, pp 215-244

[9] John Reynolds Separation Logic: A Logic for Shared Mutable Data Structures. LICS '02 Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science Pages 55-74

[10] Viktor Vafeiadis. Concurrent separation logic and operational semantics. MFPS 2011.