

Année 2018-2019 - Offre de stage Recherche - Master 2

Programmation réactive synchrone : sémantique et application à l'internet des objets

Encadrants : W. Bousdira, F. Dabrowski

Equipe d'accueil : LIFO/LMV

Thématique : Sémantique des systèmes concurrents

Laboratoire d'accueil

Le Laboratoire d'Informatique Fondamentale d'Orléans (LIFO, EA 4022) est un laboratoire de l'Université d'Orléans et de l'INSA Centre-Val de Loire. Les recherches menées au LIFO concernent la science informatique et les STIC. Elles vont de l'algorithmique au traitement des langues naturelles, de l'apprentissage au parallélisme massif, de la vérification et la certification à la sécurité des systèmes, du Big Data aux systèmes embarqués.

Equipe d'acceuil

L'objectif de l'équipe LMV est de contribuer à l'amélioration de la compréhension des problèmes de sûreté et de sécurité des systèmes informatiques. Des logiques «ordres partiels» aux langages de programmation usuels, les membres de l'équipes travaillent sur ces questions à différents niveaux d'abstraction et selon différents points de vue tout en cherchant à comprendre les relations fondamentales entre ces différentes approches.

Exposé du sujet (en langage vulgarisé) :

Le modèle de programmation réactive synchrone [9] est une relaxation du modèle du langage de programmation ESTEREL [6]. Dans ce dernier, une collection de processus progressent par succession de phases de synchronisation globale ("ticks d'horloge") et communiquent par événements. A chaque instant (phase entre deux ticks) chaque composant réagit à la présence/absence d'événements. Alors que le modèle d'ESTEREL résout le problème de causalité posé par la réaction instantanée à l'absence en restreignant l'expressivité du langage, le modèle réactif synchrone introduit la réaction retardée à l'absence (l'absence d'un signal ne peut être décidée qu'à la fin d'un instant). Plusieurs travaux ont porté sur l'étude sémantique de langages dérivés de ce modèle [2, 5, 3] et sur l'analyse des propriétés des programmes [4, 12]. D'autres travaux ont étudié avec succès la possibilité d'utiliser ce modèle de programmation dans le cadre de la mobilité [7]. Des implantations de ce modèle de programmation ont été proposées pour plusieurs langages comme C, Scheme, Ocaml et Java [8, 13, 10, 11]. Dans [9], les auteurs proposent une version du langage disposant de propriétés de sûreté fortes.

Au cours de ce stage, on s'intéressera à l'étude d'une approche purement fonctionnelle du modèle réactif synchrone. Dans un premier temps, le candidat sélectionné aura à réaliser un travail de bibliographie sur le sujet. Dans un second temps, on s'intéressera à la conception d'un micro-langage pour lequel une sémantique formelle sera réalisée en Coq[1]. Cette formalisation conduira à la réalisation d'un interprète certifié du langage. D'un point de vue appliqué l'objectif du stage est de franchir une première étape vers la proposition d'un langage de programmation de haut niveau pour l'internet des objets.

La suite naturelle de ce stage sera la préparation d'un Doctorat en informatique à l'université d'Orléans.

Références

- [1] The coq proof assistant. <https://coq.inria.fr/>.
- [2] R. M. Amadio. The sl synchronous language, revisited. *The Journal of Logic and Algebraic Programming*, 70(2):121 – 150, 2007. Algebraic Process Calculi: The First Twenty Five Years and Beyond.
- [3] R. M. Amadio. A synchronous pi-calculus. *Inf. Comput.*, 205(9):1470–1490, 2007.
- [4] R. M. Amadio and F. Dabrowski. Feasible reactivity in a synchronous pi-calculus. In *Proceedings of the 9th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, PPDP '07, pages 221–230, New York, NY, USA, 2007. ACM.
- [5] R. M. Amadio and M. Dogguy. On affine usages in signal-based communication. *CoRR*, abs/0804.1729, 2008.
- [6] G. Berry and G. Gonthier. The esternel synchronous programming language: design, semantics, implementation. *Science of Computer Programming*, 19(2):87 – 152, 1992.
- [7] G. Boudol. Ulm: A core programming model for global computing. In D. Schmidt, editor, *Programming Languages and Systems*, pages 234–248, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [8] F. Boussinot. Fairthreads: mixing cooperative and preemptive threads in c. *Concurrency and Computation: Practice and Experience*, 18:445–469, 2006.
- [9] F. Boussinot and F. Dabrowski. Safe Reactive Programming: the FunLoft Proposal. In *MULTI-PROG'08*, pages –, Sweden, 2008.
- [10] F. Boussinot and J.-F. Susini. The sugarcubes tool box: A reactive java framework. *Softw. Pract. Exper.*, 28(14):1531–1550, Dec. 1998.
- [11] L. Mandel and C. Pasteur. Réactivité des systèmes coopératifs : le cas de ReactiveML. In D. Pous and C. Tasson, editors, *JFLA - Journées francophones des langages applicatifs*, Aussois, France, Feb. 2013. Damien Pous and Christine Tasson.
- [12] A. A. Matos, G. Boudol, and I. Castellani. Typing noninterference for reactive programs. *The Journal of Logic and Algebraic Programming*, 72(2):124 – 156, 2007. Programming Language Interference and Dependence.
- [13] M. Serrano, F. Boussinot, and B. Serpette. Scheme fair threads. In *Proceedings of the 6th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, PPDP '04, pages 203–214, New York, NY, USA, 2004. ACM.

Mots-clé : Sémantique des langages de programmation, Concurrency, Assistants de Preuves, Certification, Types Linéaires

Emails des encadrants:

wadoud.bousdira@univ-orleans.fr

frederic.dabrowski@univ-orleans.fr

Candidature : Par email adressé aux deux encadrants avant le 11 Janvier 2019.