

Description formelle d'applications dans les systèmes répartis à sémantique faible

Stage de recherche proposé au LINA
par Claude Jard, Achour Mostefaoui et Matthieu Perrin

Depuis les années 1970, un gros effort a été fourni pour isoler les entités logiques d'un programme en regroupant les données et les algorithmes les traitant dans les mêmes *objets*, formellement spécifiés par des *types de données abstraits*, qui définissent à la fois les méthodes permettant de lire ou modifier les objets et des propriétés mathématiques attendues sur ces objets. Si la notion de types de données abstraits est aujourd'hui bien intégrée dans la communauté de la programmation séquentielle, elle peine à s'affirmer dans le monde de la programmation répartie. En effet, il est impossible de considérer que les méthodes d'un objet partagé entre plusieurs processus sont appelées de façon séquentielle, ce qui complique la définition de tels objets. Par exemple, si plusieurs processus tentent de modifier le même registre partagé en même temps, quelle valeur devra être retournée à la prochaine lecture ?

En fait, l'implémentation d'un tel registre n'est pas toujours possible. Dans un système réparti asynchrone où les processus communiquent en s'échangeant des messages, l'implémentation d'un registre partagé ne peut pas être obtenue sans faire l'hypothèse qu'une majorité de processus s'exécutera correctement jusqu'à la fin [1]. Depuis la démonstration de ce résultat, le modèle de *synchronisation sans attente* [2], qui suppose l'existence de registres partagés, a été largement étudié. Les *systèmes à sémantique faible*, dans lesquels on ne suppose pas la correction d'une majorité de processus, sont par contre bien moins connus, bien qu'ils soient plus appropriés pour modéliser de nombreux systèmes réels, en particulier les réseaux pair-à-pair.

Le but de ce stage est d'étudier l'expressivité de ces systèmes à travers les types de données abstraits qui y sont implémentables. On propose d'explorer la possibilité d'utiliser le langage Orc [3] développé par J. Misra à Austin pour décrire des applications utilisant la sémantique faible.

Ce stage s'adresse à des étudiants motivés par la formation par la recherche, intéressés par les systèmes répartis et les méthodes formelles.

Références

- [1] Hagit Attiya, Amotz Bar-Noy, and Danny Dolev. Sharing memory robustly in message-passing systems. In *Journal of the ACM*, 1995.
- [2] Maurice P. Herlihy. Impossibility and universality results for wait-free synchronization. In *Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, PODC '88, pages 276–290, New York, NY, USA, 1988. ACM.
- [3] David Kitchin, Adrian Quark, William Cook, and Jayadev Misra. The orc programming language. In David Lee, Antónia Lopes, and Arnd Poetsch-Heffter, editors, *Formal Techniques for Distributed Systems*, volume 5522 of *Lecture Notes in Computer Science*, pages 1–25. Springer Berlin / Heidelberg, 2009.