

Types liquides pour les fonctions de flot

Jean-Pierre.Talpin@inria.fr

Un système cyber-physique est une entité aux constituants hétérogènes: c'est un logiciel, embarqué sur un matériel, interfacé avec le monde physique. Le temps prend des formes différentes observé d'un de ces points de vue ou d'un autre: il est continu en physique, événementiel en logiciel, déclenché par horloge dans le matériel. Plus encore, les paradigmes de modélisation et de programmation utilisés pour le logiciel (synchrone), le matériel (RTL, TLM) ou sa physique (EDO) en modifient encore la perception et la compréhension. Pourtant, les contraintes temporelles d'un système doivent non seulement concorder entre ses constituant, mais aussi en cohérence avec les autres quanta qui en dépendent: vitesse, fréquence, puissance, débit, donc taille, volume, pression, capacité, chaleur, angle, etc.

À la lumière des progrès récents en théorie de type et en vérification formelle, avec la définition des classes décidables de systèmes de type dépendant, le développement de fonctionnalités de vérification SAT/SMT dans les outils de démonstration automatique, notre projet est de définir un langage flot de données fortement typé capable de représenter et d'analyser chaque aspect logique et quantitatif du comportement d'un programme et de son interface de communication avec l'environnement d'exécution physique.

Notre objectif est de concevoir et d'appliquer un cadre de programmation basé sur la théorie des types dépendants pour analyser, vérifier, tracer, valider, certifier formellement les spécifications de systèmes temps-réel et leurs mises en œuvre.

Le programme de thèse consistera à la fois en élaboration de fondement théorique et en la mise en œuvre d'un prototype. On pourra pour cela s'aider de l'infrastructure du projet POP Eclipse (Polychrony sur Polarsys). Il s'adresse à un étudiant motivé avec une solide expérience en informatique théorique, et, plus spécifiquement, langages de programmation, compilation, théorie des types, vérification formelle, logique en programmation. Une connaissance pratique ou une expérience passée avec l'un des sujets ci-dessus, ou bien avec Haskell, Coq, OCaml, sont autant d'atouts, de même qu'un anglais courant, parlé et écrit.

Références

"Liquid types for stream-processing functions". J.-P. Talpin, et al. Draft manuscript, 2015.

"The type and effect discipline". Talpin, J.-P., Jouvelot, P. Conference on Logic in Computer Science (LICS'92). IEEE Press, June 1992. ACM-IEEE Test-of-Time Award 2012.

"Implementation of the typed lambda calculus using a stack of regions". Tofte, M., Talpin, J.-P. Symposium on Principles of Programming Languages (POPL'94). ACM Press, January 1994. ACM SIGPLAN most influential POPL paper Award 2004.

