INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
INRIA
centre de recherche SACLAY - ÎLE-DE-FRANCE

LABORATOIRE DE RECHERCHE EN INFORMATIQUE

Proposition de Stage

Preuve de programmes avec lieurs

Lieu

Equipe-Projet ProVal
http://www.lri.fr/proval
Batiment 650
Rue Noetzlin
Université Paris-Sud
Orsay, France

Personnes encadrant le stage

Claude Marché (Equipe ProVal)
Tél : 01 72 92 59 69
Email : Claude.Marche@inria.fr

Andrei Paskevich (Equipe ProVal)
Tél : 01 74 85 42 82
Email : Andrei.Paskevich@lri.fr

Description du travail de stage

Why3 (http://why3.lri.fr [4]) est un environnement pour la preuve de programmes. Il propose un langage de programmation ad-hoc qui permet de coder mais aussi de spécifier le comportement des programmes. Son générateur d'obligations de preuves produit des formules logiques (en logique du premier ordre), qui doivent être prouvées valides par des outils de preuve externes, aussi bien des prouveurs automatiques comme Alt-Ergo (http://alt-ergo.lri.fr [6]), CVC3 (http://cs.nyu.edu/acsys/cvc3/ [2]) ou Z3 (http://research.microsoft.com/en-us/um/redmond/projects/z3/ [5]), que interactif comme Coq (http://coq.inria.fr/ [3]). La validité de ces formules implique que le programme respecte ses spécifications. Des exemples de programmes spécifiés et prouvés en Why3 se trouvent sur la galerie Web http://proval.lri.fr/gallery/why3.en.html.

Un objectif pour le projet Why3 est d'être capable de prouver des programmes qui manipulent des expressions symboliques (prouveur, compilateur, etc.). Une difficulté majeure dans un tel contexte est de devoir travailler sur des types de données avec lieurs. Ce sujet fait l'objet d'un défi international appelé POPLmark (https://alliance.seas.upenn.edu/~plclub/cgi-bin/poplmark/ [1]). Plusieurs solutions ont déjà été proposées, en particulier basées sur des assistants de preuve comme Coq ou Isabelle, qui proposent des langages de spécification évolués, avec en particulier des formules d'ordre supérieur.

L'objet du stage sera de développer en Why3 des programmes manipulant des lieurs. Les difficultés à surmonter seront de deux ordres. D'une part, le langage de spécification étant plus pauvre que ceux des assistants ci-dessus, il faudra identifier une façon nouvelle de spécifier les lieurs. D'autre part, il s'agira de formuler les spécifications d'une manière qui permette un maximum d'automatisation des preuves, et donc un minimum de travail de preuve interactive.

Le travail devra être validé par des études de cas de complexité croissante. Un premier sujet d'étude sera le lambda-calcul pur. On s'intéressera aussi à formaliser la logique du premier ordre, l'unification, la preuve par résolution. Si le temps le permet, le défi POPLmark (Système F avec sous-typage) sera étudié aussi. On s'attachera à essayer de mettre en œuvre une méthodologie commune, qui pourra se matérialiser par exemple par des théories génériques, et donc réutilisables, des lieurs.

Prérequis souhaités

Connaissances de base en logique, théorie de la démonstration et en méthodes de spécification formelle de programmes. Une expérience de programmation en Objective Caml serait un plus apprécié.

Références

[1] B. Aydemir, A. Bohannon, M. Fairbairn, J. Foster, B. Pierce, P. Sewell, D. Vytiniotis, G. Washburn, S. Weirich, and S. Zdancewic. Mechanized metatheory for the masses : The POPLmark challenge. In *Proceedings of the Eighteenth International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2005)*, number 3603 in Lecture Notes in Computer Science, pages 50–65. Springer, 2005.

[2] Clark Barrett and Cesare Tinelli. CVC3. In Werner Damm and Holger Hermanns, editors, *19th International Conference on Computer Aided Verification*, volume 4590 of *Lecture Notes in Computer Science*, pages 298–302, Berlin, Germany, July 2007. Springer.

[3] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development.* Springer-Verlag, 2004.

[4] François Bobot, Jean-Christophe Filliâtre, Claude Marché, and Andrei Paskevich. Why3 : Shepherd your herd of provers. In *Boogie 2011 : First International Workshop on Intermediate Verification Languages*, Wrocław, Poland, August 2011.

[5] Leonardo de Moura and Nikolaj Bjørner. Z3, an efficient SMT solver. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[6] Stéphane Lescuyer. *Formalisation et développement d'une tactique réflexive pour la démonstration automatique en Coq.* Thèse de doctorat, Université Paris-Sud, 2011.