Acceleration de boucles et interpolation de Craig pour la vérification de programmes numeriques

Responsables de stage :
Marius Bozga, CNRS (Marius.Bozga@imag.fr)
Radu Iosif, CNRS (Radu.Iosif@imag.fr)

Laboratoire d'accueil : VERIMAG (UMR 5104)
Centre Equation, 2 avenue de Vignate, 38610 GIERES
`http://www-verimag.imag.fr`

Durée du stage : 3 à 4 mois

1 Motivation et contexte

La vérification des algorithmes, programmes et systèmes informatiques a été déjà posée au début du 20ème siècle comme une problématique de recherche par Alan Turing (1912-1954), un des fondateurs de l'informatique. Cette problématique a depuis connu des avancées significatives. A titre d'exemple, les travaux de J. Sifakis, A. Emerson, E. Clarke et A. Pnueli sur la vérification algorithmique des programmes et les liens entre la vérification des programmes et les logiques temporelles ont été récompensées (en 1996 et 2007) avec le Prix Turing, équivalent du Prix Nobel pour l'informatique.

La vérification de logiciels embarqués est actuellement un des thèmes de recherche majeurs dans le domaine des technologies du logiciel. La complexité grandissante et le caractère souvent critique de ces systèmes rend nécessaire le développement de méthodes et d'outils de conception et de validation permettant de garantir leur sûreté de fonctionnement. Dans ce contexte, le laboratoire VERIMAG se concentre sur le développement des outils ainsi théoriques que pratiques pour la vérification automatique de systèmes embarqués.

2 Vérification de programmes numeriques

Les programmes numeriques (parfois appelles systémes à compteurs ou automates à compteurs) sont des modeles de calcul d'une grande expressivité (Turing-complétes), qui encode facilement de programmes avec des structures de données d'ordre superieur (vecteurs, listes, etc.). La vérification pour ce type de programmes est un domaine trés large, à la fois théorique (on compte des importants résultats de décidabilité et de complexité, comme par exemple, les systèmes reversal-bornées ou les reseaux de Petri) ainsi que pratique ; en effet, un nombre important d'outils d'analyse ont été developpés recemment : ASTREE, INTERPROC, ARMC, FAST, FLATA, etc.

Une technique trés prometteuse est l'abstraction par predicats (Predicate Abstraction) [1], combinée avec le raffinement par contre-exemple (Counterexample Guided Abstraction Refinement) [2]. Une importante avancée dans cette direction à été l'application de l'interpolation de Craig au raffinement par contre-exemple [3]. Cette methode calcule, pour une formule insatisfaisable $A \wedge B = \bot$ un interpolant I, tel que $A \rightarrow I$ et $I \wedge B = \bot$. Souvent l'interpolant est une formule simplifiée, qui donne la raison pour laquelle la conjonction $A \wedge B$ est fausse.

Une autre technique importante de vérification est l'acceleration de boucles [4, 5], qui consiste à calculer, de maniere precise, l'effet de l'iteration d'une boucle du programme un nombre arbitraire de fois, ou, autrement dit, calculer la clôture transitive de sa relation de transition.

3 But du stage

Lors de ce stage, on cherchera dans un prémier temps, une méthode qui combine les avantages de l'interpolation de Craig (trouver des arguments simples pour l'insatisfaisabilit'e) avec les avantages de l'acceleration (génération de contre-exemples generalisés). En particulier, on essayera d'etendre les techniques d'interpolation lineaires a des theories plus puissantes (arithmetique de Presburger, ou des extensions de celle-ci).

Dans un deuxieme temps, le stagiaire aura la possibilité d'intégrer la méthode combinée au sein de l'outil FLATA, [6] un outil pour l'analyse de programmes entiers, développe a présent a VERIMAG.

Références

[1] S.Graf, H.Saïdi. Verifying Invariants Using Theorem Proving. In Proceedings of the 8th Conference on Computer-Aided Verification (CAV'96) , Rutgers University, New Jersey, July 1996.

[2] Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu and Helmut Veith. Counterexample-Guided Abstraction Refinement, Computer Aided Verification Lecture Notes in Computer Science, 2000, Volume 1855/2000, 154-169

[3] Ranjit Jhala, Kenneth L. McMillan : Lazy Abstraction with Interpolants. CAV 2006, 123-136

[4] Sébastien Bardin, Alain Finkel, Jérôme Leroux, Laure Petrucci : FAST : Fast Acceleration of Symbolic Transition Systems. CAV 2003 : 118-121

[5] Marius Bozga, Radu Iosif, Filip Konecný : Fast Acceleration of Ultimately Periodic Relations. CAV 2010 : 227-242

[6] FLATA homepage : `http://www-verimag.imag.fr/FLATA.html`