

# Preuves de terminaison pour les systemes à compteurs

## 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](http://www-verimag.imag.fr)

Durée du stage : 2 à 3 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.

Il est à noter que Grenoble a toujours joué un rôle de premier plan dans la recherche sur le problème de la vérification des algorithmes, programmes et systèmes informatiques. Le problème de la vérification des systèmes informatiques tout en ayant une pertinence indéniable dans beaucoup de secteurs informatiques industriels (cf. les investissements de plusieurs grands industriels) reste un domaine de recherche fascinant et très dynamique (cf. les recrutements au CNRS et à l'INRIA ces dernières années).

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 Systèmes à compteurs

Le modèle d'automate fini étendu avec des compteurs entiers a connu un grand succès notamment dans le domaine de l'analyse statique, où un nombre de techniques ont été proposées pour évaluer de façon automatique, les contraintes linéaires entre les variables d'un programme. L'inconvénient majeur de cette famille de techniques est l'utilisation de l'approximation, qui peut parfois rendre les résultats des analyses peu convaincants.

La vérification exacte des automates à compteurs est, en général, impossible, car les automates avec deux compteurs ont le même pouvoir expressif que les machines de Turing. En revanche, des recherches menées pendant les derniers 30 ans ont montré l'existence de classes non-triviales d'automates pour lesquelles les problèmes de vérification sont décidables, par exemple : les automates reversal-bornées, les automates plates, certaines classes d'automates avec transitions affines, etc.

### 3 But du stage

Lors de ce stage on étudiera en premier les techniques existantes pour prouver la terminaison des systèmes à compteurs (ou équivalent, des programmes non-déterministes avec des variables entières). Ici on peut noter les travaux sur les *fonctions de rang* [1, 2] et les *invariants de transitions* [3].

Ensuite, on étudiera le problème de terminaison pour une boucle de programme étiquetée par des transitions affines. Ce problème est connu comme très difficile, car il traduit un autre problème ouvert [4]. Cependant, pour des sous-classes de relations affines (contraintes de différence, contraintes octogonales), ce problème est probablement décidable.

Finalement, le stagiaire proposera des méthodes pour étendre les résultats obtenus pour une seule boucle à des systèmes complexes (avec des boucles imbriquées). Il aura également la possibilité d'implémenter les techniques élaborées au sein de l'outil FLATA, [5] développe à présent à VERIMAG.

### Références

- [1] Andreas Podelski and Andrey Rybalchenko. A complete method for the synthesis of linear ranking functions, In Proc. Verification Model Checking and Abstract Interpretation (VMCAI) 2004
- [2] Aaron R. Bradley, Zohar Manna and Henny B. Sipma, The Polyranking Principle, In Proc. International Colloquium on Automata, Languages and Programming (ICALP) 2005, Lecture Notes in Computer Science, Volume 3580,
- [3] Andreas Podelski and Andrey Rybalchenko. Transition Invariants. In Proc. Logic in Computer Science (LICS) 2004.
- [4] Terence Tao. Open question : effective Skolem-Mahler-Lech theorem.  
<http://terrytao.wordpress.com/2007/05/25/open-question-effective-skolem-mahler-lech-theorem/>
- [5] FLATA homepage : <http://www-verimag.imag.fr/FLATA.html>