Analyse Statique de Programmes sous Contraintes de Temps

Raphaël Monat*

raphael.monat@inria.fr rmonat.fr

Contexte

L'analyse statique permet de prouver automatiquement des propriétés sur des programmes, par exemple pour détecter des erreurs à l'exécution.

L'idée est d'interpréter le programme dans un domaine spécifique permettant de calculer en temps fini une surapproximation de l'ensemble des états de programmes atteignables.

Le cadre de l'interprétation abstraite [5] permet de formaliser cette notion d'interprétation des programmes, et de prouver que l'interprétation est sûre vis-à-vis de la sémantique du programme original, afin de s'assurer qu'aucun cas d'exécution n'a pas été pris en compte.

But

Les analyses statiques par interprétation abstraite sont souvent garanties de terminer en temps fini. Néanmoins, il serait intéressant de pouvoir spécifier des contraintes de ressources (temps CPU, mémoire) qu'un analyseur statique devrait respecter en adaptant sa précision. Un cas d'application immédiat est la compétition de vérification de logiciels SV-Comp [3], qui limite chaque tache de vérification par 15 minutes de temps CPU et 8 Go de mémoire. Nous avons participé à la compétition cette année avec la soumission de l'outil Mopsa [4], et remporté une médaille de bronze dans une catégorie.

Le but de ce stage est de développer des techniques permettant d'estimer combien de temps une analyse statique donnée prendra sur un programme spécifié. Un point de départ pourrait être le développement d'une pré-analyse qui pourrait estimer la complexité de l'analyse de programme via une formule symbolique (Ballabriga et al [2]).

Thématique

Les thèmes abordés dans ce stage sont fortement liés aux notions de sémantique et d'interprétation des programmes.

Localisation

Le stage aura lieu dans l'équipe Inria SyCoMoRES, au sein du laboratoire CRIStAL près de Lille. Lille est une ville étudiante, proche de Bruxelles, Paris et Londres, facilement accessible en train. Les températures maximales de l'été sont plutôt supportables.

*N'hésitez pas à ma contacter si vous avez des questions ou aimeriez discuter d'un autre sujet de stage !

Références

1. A Progress Bar for Static Analyzers, Lee, Oh, Yi
2. Symbolic WCET computation, Ballabriga, Forget, Lipari
3. Competition on Software Verification
4. The Mopsa Static Analysis Platform
5. Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation