Analyse statique de programmes

Encadrant : Sujet proposé par David Monniaux, directeur de recherche au CNRS.

Lieu : laboratoire VERIMAG, Grenoble.

1 Cadre

L'analyse statique de programmes consiste à calculer automatiquement des propriétés sur les exécutions possibles du programme passé en entrée à l'analyseur ; on pourra par exemple vouloir démontrer automatiquement que ce programme ne fait jamais d'accès en dehors des bornes de tableau, ou vérifier des propriétés arbitraires spécifiées par l'utilisateur (`assert` en langage C). Bien entendu, il s'agit en toute généralité d'une tâche impossible, car de telles propriétés sont indécidables ; toutefois, on arrive en pratique à décider de nombreuses propriétés utiles. [1]

Le procédé que nous utilisons, l'interprétation abstraite, consiste à calculer des sur-ensembles des états accessibles du programme en les différents points de son flot de contrôle.

2 Sujet

Notre équipe dispose déjà d'un outil d'analyse statique (PAGAI), mais celui-ci, développé en C++, est peu pratique pour apporter des modifications et expérimenter.

Le stagiaire devra donc implanter un outil d'analyse statique en Objective Caml ou F#, prenant en entrée du bitcode LLVM (on utilisera les bibliothèques de lecture de bitcode LLVM ; on pourra éventuellement remplacer LLVM par la lecture de bytecode Java ou .net, suivant les compétences et affinités du stagiaire), et s'appuyant sur des outils existants de SMT-solving (par exemple Z3) et de calcul polyédral (par exemple libpoly, développée dans l'équipe).

3 Compétences requises

Logique mathématique.
Programmation en langage Objective Caml et/ou F#.

1. L'encadrant du stage a auparavant travaillé à l'ENS et y a été co-développeur de l'outil Astrée, qui est depuis commercialisé et utilisé notamment par des industriels de l'avionique.