Encadrant : Nicolas Stouls

Contact : nicolas.stouls@insa-lyon.fr

Titre : Assistance à la vérification automatique de propriétés dynamiques sur du code C

Contexte : *Preuve de programme*
La preuve de programmes est une méthode de vérification formelle consistant à établir, par la preuve, que toute exécution d'un programme respecte une propriété donnée. Par exemple, la chaine d'outils Frama-C/Jessie/Why[1] permet de générer des obligations de preuves[2] à partir d'un programme C contenant des propriétés sous la forme d'annotations (décrite avec le langage d'annotations ACSL). Classiquement, la preuve de programmes se limite aux propriétés invariantes. Cependant de récents travaux [TH02,Gro07,GS09] ont proposé différentes méthodes d'encodage de propriétés dynamiques sous la forme d'invariants. Bien que cette traduction soit nécessairement partielle, nous proposons des heuristiques permettant de favoriser la faisabilité et l'automaticité des obligations de preuve générées. Ce stage s'insère dans la réalisation de l'outil Aoraï [GS09], qui est un greffon de la plateforme Frama-C pour la preuve de programmes C. Durant ce stage, l'étudiant pourra être amené à communiquer avec les développeurs de la chaine d'outils utilisée (CEA, INRIA) ou des utilisateurs du programme développé (Dassault aviation, Airbus, Atos Origin, ...)

Sujet
L'objectif de ce stage est d'ajouter de nouvelles fonctionnalités au logiciel Aoraï. Plusieurs ont déjà été demandées pour les utilisateurs potentiels et ce sera au stagiaire de choisir les points qu'il souhaite traiter en premier. Parmi les améliorations attendues, nous trouvons :

- Mise à jour du langage de spécification
- Ajout d'une heuristique considérant des automates déterministes
- Amélioration de la lisibilité des spécifications générées
- Implémentation des variables ghost
- etc.

Dans un premier temps, le candidat manipulera l'outil Aoraï pour mieux comprendre son fonctionnement et les choix effectués. Ensuite, à partir du cahier des charges, le candidat sera amené à proposer ses propres solutions algorithmiques de mise à jour. Une réflexion pourra également être menée sur les extensions possibles du langage de spécification ACSL, pour y inclure des primitives comportementales.

Bibliographie
[Gro07] J. Groslambert. Vérification de propriétés temporelles par génération d'annotations. PhD thesis, Université de Franche-Comté, 2007.

[GS09] Julien Groslambert et Nicolas Stouls, *Vérification de propriétés LTL sur des programmes C par génération d'annotations*, in *AFADL'09*.

[TH02] K. Trentelman and M. Huisman. *Extending JML Specifications with Temporal Logic.* In AMAST'02, number 2422 in LNCS, pages 334–348.

1 http://www.frama-c.cea.fr/
2 Une obligation de preuve est une formule logique dont on cherche à établir la verracité.