Proposition de stage Pro niveau bac+5

Monitoring optimisé pour la détection des erreurs de mémoire dans les programmes C

Mots-clés : allocation dynamique, validité des pointeurs, vérification des programmes C, spécification formelle, *runtime assertion checking*

Cadre

Le CEA LIST est un centre de recherche technologique sur les systèmes à logiciel prépondérant qui mène ses recherches en partenariat avec les grands acteurs industriels du nucléaire, de l'automobile, de l'aéronautique, de la défense et du médical pour étudier et développer des solutions innovantes adaptées à leurs besoins. Au sein du CEA LIST, le Laboratoire de Sûreté des Logiciels (LSL), localisé à Saclay (Essonne, 91), développe des outils d'aide à la validation et à la vérification de logiciels et de systèmes matériels/logiciels, tout particulièrement dans le domaine des systèmes embarqués critiques.

L'un des nos outils, nommé Frama-C (`http://frama-c.com`), est une plate-forme logicielle facilitant le développement d'outils d'analyses de programmes C. Le stage se déroulera au sein de l'équipe de R&D développant Frama-C.

Objectifs

Chaque programme C analysé par Frama-C peut être annoté par des spécifications formelles, écrites dans un langage appelé ACSL [1]. Frama-C offre alors différentes techniques de vérification pour garantir que le programme satisfait sa spécification. Une des techniques a pour but de traduire une sous-classe des annotations ACSL– celles dites exécutables – en instructions C intégrées au programme sous analyse [2]. Cette transformation permet d'obtenir un nouveau programme C dont la correction vis-à-vis de sa spécification est vérifiée dynamiquement, pendant son exécution : cette technique est appelée le *runtime assertion checking*. Une des difficultés principales de cette transformation réside dans la prise en compte du modèle mémoire du langage C. Par exemple, un accès à un tableau hors limites (e.g. avec un indice trop grand), ou à une zone mémoire allouée dynamiquement et ensuite libérée, serait invalide en C. Une bibliothèque (env. 1000 lignes de code C) a été développée [3] pour collecter les allocations, dé-allocations et initialisations effectuées par le programme C et contrôler ensuite la validité (et d'autres propriétés) des accès mémoires.

Ce stage vise à développer une extension de la bibliothèque qui intégrera de nouvelles fonctionnalités pour une meilleure détection de certaines erreurs. Notamment, des tentatives d'utilisation d'une zone mémoire libérée et ré-allouée à nouveau, ou des décalages de pointeur dans un autre bloc en dehors du bloc mémoire initial, ou des accès à cheval entre deux blocs mémoire seront pris en compte. Un deuxième axe des travaux serait l'intégration des techniques de monitoring récentes consistant à surveiller la validité de la mémoire grâce à une copie (*shadow page*) avec des accès fortement optimisés qui pourront améliorer les performances de la bibliothèques. Ce stage sera l'occasion d'acquérir une bonne expérience de développement pointu en C ainsi qu'une expertise en gestion de la mémoire et détection des anomalies.

Candidatures

Une bonne maîtrise du langage C, notamment en gestion de la mémoire. Connaissances en vérification de programmes seraient un plus. Les délais administratifs de recrutement au CEA étant de 2 à 3 mois minimum, merci de prendre contact le plus tôt possible.

Contacts : Nikolai Kosmatov et Julien Signoles (`prenom.nom@cea.fr`)

Références

[1] P. Baudin, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, and V. Prevosto. *ACSL : ANSI/ISO C Specification Language, version 1.7*, 2013. `http://frama-c.com/acsl.html`.

[2] M. Delahaye, N. Kosmatov, and J. Signoles. Common specification language for static and dynamic analysis of C programs. In *Symposium on Applied Computing (SAC'13)*, pages 1230–1235, 2013.

[3] N. Kosmatov, G. Petiot, and J. Signoles. An optimized memory monitoring for runtime assertion checking of C programs. In *International Conference on Runtime Verification (RV 2013)*, pages 167–182, 2013.