Visit of Xavier Leroy

Schedule for October 20th, 2015

10:00-12:00 Visit of the laboratory

Meeting the following teams:

10:00 - 10:30 Meeting with Frédéric Peschanski (Apr team)

10:30 - 11:00 Meeting with Karine Heydemann (Alsoc team)

11:00 - 11:30 Meeting with Marc Shapiro (Regal team)

11:30 - 12:00 Meeting with Julia Lawall (Whisper team)

12:00-14:00 Lunch

14:00-16:00 Masterclass (room 24-25/405)

Short presentations by the PhD students. Each presentation ($\approx$ 10 minutes) will be followed by an open discussion with Xavier Leroy ($\approx$ 10 minutes).

- Loïc Girault (Move team)

  Title: Réingénierie d'architecture via les graphes de dépendances

  Supervisors: Cédric Besse, Mikal Ziane

  Abstract: Une architecture doit être modulaire : supprimer les dépendances indésirables est essentiel lors de la réingénierie d'une architecture. On vise à réduire le coût de certains changements en empêchant leur propagation. On part du principe qu'utiliser une structure relationnelle permet de raisonner sur les dépendances plus simplement que sur du code. Nous développons une méthode permettant de travailler à ce niveau d'abstraction. On construit à partir d'un AST un graphe de dépendances. On identifie les dépendances indésirables sur ce graphe et on applique une séquence d'opérations permettant de les supprimer tout en préservant le comportement. Pour terminer on concrétise cette séquence de transformations faite sur le graphe sur l'AST qui lui même est imprimé pour donner le code réingénieré. Nous réalisons cette approche au travers d'un prototype, Puck.

- Bruno Ezvan (Alsoc team / Trusted labs)

  Title: Vérification formelle de circuits sécurisés

  Supervisors: Karine Heydemann, Emmanuelle Encrenaz

  Abstract: Ma recherche s'effectue à la suite de travaux de modélisation d'un circuit numérique décrit en Verilog, un langage de description matériel, au sein de l'assistant de preuve Coq à des fins de certification. Les circuits numériques ayant des fonctionnalités sensibles peuvent être objet d'attaques physiques comme l'injection de fautes. Je propose la formalisation d'une attaque par injection de fautes ajoutée au modèle Coq du circuit. Cette formalisation m'a permis de vérifier formellement que la contre-mesure implémentée dans le circuit est efficace considérant ce modèle d'attaquant. Dans un second temps, j'ai développé un outil générant un modèle Coq du circuit à partir d'une description en langage Verilog. Nous avons adapté notre modèle pour permettre la vérification en prouvant individuellement la spécification formelle de chaque module que nous combinerons pour vérifier l'ensemble du circuit. Nous avons utilisé cette approche pour vérifier une implémentation matérielle de la primitive cryptographique AES (Advanced Encryption Standard).

- Mahsa Najafzadeh (Regal team)

  Title: Verifying and co-designing program semantics

  Supervisor: Marc Shapiro

  Abstract: Reasoning about concurrent updates in a replicated database is a difficult task because operations might execute in different orders at different replicas. In this work, we formalize and verify a distributed program with concurrent users. We derive a set of sound proof rules, called CISE, that allow the verification of distributed programs in polynomial time. A CISE-enabled tool verifies whether the concurrent execution of a given program will maintain its specific correctness invariants under a minimal consistency model, if not, the tool provides a counter-example, which guides the program developer to understand which concurrent executions cause conflict. To address conflicts, the program developer has a choice of either strengthening the consistency protocol or weakening the program semantics.

- Vincent Botbol (Apr team / Cea)

  Title: Verification of distributed systems using symbolic transducers

  Supervisors: Emmanuel Chailloux, Tristan Le Gall

  Abstract: Performing static analysis on distributed systems is often a challenging task. To achieve this goal, we present a concurrency model based on symbolic transducers. In order to prove safety properties on such a model, one needs to solve a reachability problem,

which is undecidable. We use the abstract interpretation framework to compute an over-approximation of the reachability set, so we can either prove the safety property or raise an alarm flag.

- Steven Varoumas (Apr team)

  Title: Programmation concurrente de microcontrôleurs basée sur une approche machine virtuelle

  Supervisor: Philippe Trébuchet, Tristan Crolard

  Abstract: À partir de la machine virtuelle OCaPIC permettant l'exécution de code-octet OCaml sur des microcontrôleurs à faibles ressources, on propose un modèle de programmation concurrente adapté à la nature des systèmes embarqués. Celui-ci, basé sur le langage de programmation synchrone à flots de données Lustre, prend la forme d'OCaLustre : un prototype d'extension synchrone du langage OCaml. On aborde en particulier le modèle de compilation d'OCaLustre, dont le principal objectif est de proposer une consommation mémoire la plus limitée possible.