INRIA — Département d'Informatique de l'École Normale Supérieure

Automatic Verification of an Abstract Machine for Synthesized Operating System Schedulers

Internship location: École Normale Supérieure ; 45, rue d'Ulm ; 75 230, PARIS.

Team: Équipe Sémantique et Interprétation Abstraite / Équipe-Projet "ANTIQUE".

Advisor & Contact : Xavier RIVAL (*e-mail* : `rival@di.ens.fr`, tél : 01 44 32 21 50, fax : 01 44 32 21 51)

Internship topic:

The scheduler of an Operating System such as Linux is a very complex piece of software, aimed at controlling the order in which processes access resources such as CPU time or I/O. The Linux scheduler implements a highly parametric policy, that can be used on a wide range of architectures (from micro-controllers to servers). In many cases such as embedded systems, scheduling is a very critical task and programming errors may lead to starvation (some task does not execute anymore) or to performance bugs. In this context, it is important to verify, if possible formally task preservation, and the absence of starvation.

The *domain-specific language* (DSL) approach consists in designing programming languages that provide abstractions tailored to a given application. It has been successfully applied to the synthesis of schedulers from a concise and more readable specification [1]. Programmers may then focus on the characteristics that are specific to the scheduling policy and let the DSL compiler generate optimized C code, to be run on top of a small abstract machine, crafted by expert developers. This means that implementing schedulers can be made less error prone and that maintenance and adaptation of the scheduling policies is also simplified.

Moreover, this approach opens up an opportunity for the synthesis of *verified schedulers*. Indeed, while the verification of hand-written C code is a major challenge, a framework based on a DSL is much more amenable to verification, since one may adapt the compiler to generate code that is easier to analyze successfully, using automatic techniques such as abstract interpretation based static analysis.

This internship aims at setting up the foundations for a verified DSL-based scheduler compilation framework, based on the **MemCAD** [2,3,4,5] generic abstract interpreter for programs manipulating complex data-structures. To achieve this, the internship will consider the following tasks:

1. **Formalization of the verification target properties**. Target properties should include memory safety, the preservation of all the data-structures used by the scheduler and the robustness of the complete scheduler (it should not lose tasks or leave some resources unused when tasks are waiting). To formalize these properties, the **MemCAD** structure specification language will need to be extended.
2. **Design of domain specific abstract domains**. To verify these target properties, the **MemCAD** static analyzer needs to be equiped with adequate abstract domains, that is, predicate algebra with efficient machine representation and reasoning algorithms. While the basic data-structures based on lists or trees can already be handled by **MemCAD**, advanced data-structure properties specific to the scheduling problem cannot. These include the sortedness of run-queues, or the balancing of tree data-structures. Therefore, new abstract domains will be need to be designed, formalized and proved correct.
3. **Implementation of the verifier**. Finally, a verifier based on the **MemCAD** tool will be implemented so as to verify a basic implementation of a DSL-generated scheduler, provided by a partner team specialized in DSLs.

Funding: This project is part of the VeriAMOS ANR project. **PhD funding is available**.

References

[1] A Framework for Simplifying the Development of Kernel Schedulers: Design and Performance Evaluation. Gilles Muller, Julia Lawall, and Herve Duchesne. In HASE'05 (International Symposium on High Assurance Systems Engineering Conference), pages 56–65, 2005.

[2] Dino Distefano, Peter W. O'Hearn, Hongseok Yang. A Local Shape Analysis Based on Separation Logic. In TACAS'06, pages 287-302, 2006.

[3] Bor-Yuh Evan Chang and Xavier Rival. Relational inductive shape analysis. In POPL'08, pages 247–260, 2008.

[4] Huisong Li, Francois Berenger, Bor-Yuh Evan Chang, Xavier Rival. Semantic-directed clumping of disjunctive abstract states. In POPL'17, pages 32–45, 2017.

[5] MemCAD static analyzer. `https://www.di.ens.fr/~rival/memcad.html`