Exact Register Allocation in the Jasmin Certified Compiler

Keywords Compilation, algorithms

City and country Nancy, France
Team and research laboratory Team Pesto, at Loria lab (Inria Nancy, CNRS, Université de Lorraine)
Name and address of the advisor Vincent Laporte, Vincent.Laporte@inria.fr
Name and address of the head of the laboratory Yannick Toussaint, yannick.toussaint@loria.fr

Funding The internship is supported by the PÉPR "Cybersécurité" (France 2030 program managed by the French National Research Agency under grant agreement No. ANR-22-PECY-0006).

Context

Jasmin is a low-level programming language supporting the development of high-assurance cryptography libraries (Almeida et al. 2017). This language features an equilibrium between the high-level abstractions that ease the implementation of programs and reasoning about them on one hand, and on the other hand access to low-level features together with the predictability of the compiler that give control on performances to the programmer.

The Jasmin infrastructure has been used to provide formal security guaranties on efficient implementations of the SHA-3 hash function (Almeida et al. 2019) and of the ML-KEM post-quantum primitive (Almeida et al. 2023). The correctness proof of the compiler (mechanized in the Coq proof assistant) justifies the formal reasoning done at the source level.

Objective of the internship

Register allocation is one on the numerous passes performed by the Jasmin compiler. In accordance with the design principles of the language, this compilation pass never introduces any spilling operation. In its current implementation, the compiler translates the register allocation problem into a NP-hard graph coloring problem and attempt at solving it (in polynomial time) using an approximate algorithm. The aim of the internship is to study the opportunity of using an exact algorithm for optimal register allocation in the Jasmin certified compiler. To this end two lines of work shall be explored.

A first family of algorithms exploit the structure of control-flow graphs, such as their small tree-width or other grammatical decompositions. Indeed, when restricted to specific classes of graphs, register allocation may be solved in polynomial time (Thorup 1998). In spite of their appealing asymptotic complexity, these algorithms often have a high cost in practice and have been so far restricted to target architectures with few registers. However, recent works suggest that these algorithms may be practical, even when the target architecture features many registers (Conrado, Goharshady, and Lam 2023). Since by design of the programming language, Jasmin programs are structured and the compiler should not introduce any spill, these algorithms appear to be well suited there.

A second approach to optimal register allocation rely on constraint solving: the allocation problem is encoded as a set of constraints that are then solved by an off-the-shelf solver. This methodology has shown to be successful even for other compilation passes such as instruction scheduling (Lozano et al. 2019). Bringing the expressivity of constraint programming into the Jasmin compiler —in addition to immediate impact on register allocation— would pave the way to generically solve harder problems such as aliasing of registers in Neon architectures.

More precisely, the intern will:

- study two algorithms for register allocation: one that exploit the structure inherent to control-flow graphs, and one that builds up on constraint programming;
- adapt them to the Jasmin programming language and implement them in its compiler;
- compare them and evaluate them experimentally both in terms of compile-time performance and quality of the generated code.

Expected ability of the student

We expect mathematical maturity, basic knowledge in logic, basic theoretical computer science, some understanding of compilation. Knowledge in security and cryptography is not mandatory. Ability to program in OCaml would be appreciated.

Note that there is funding available to extend this internship towards a PhD, would the candidate be interested.

Bibliographic references

Almeida, J. B., Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Vincent Laporte, Jean-Christophe Léchenet, Tiago Oliveira, et al. 2023. "Formally verifying Kyber: Episode IV: Implementation correctness." In *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023:164–93. 3. Praha, Czech Republic: IACR. https://doi.org/10.46586/tches.v2023.i3.164-193.

Almeida, J. B., M. Barbosa, G. Barthe, A. Blot, B. Grégoire, V. Laporte, T. Oliveira, H. Pacheco, B. Schmidt, and P.-Y. Strub. 2017. "Jasmin: High-Assurance and High-Speed Cryptography." In *Proceedings of the 24th ACM Conference on Computer and Communications Security,* (*CCS*).

Almeida, J. B., C. Baritel-Ruet, M. Barbosa, G. Barthe, F. Dupressoir, B. Grégoire, V. Laporte, T. Oliveira, A. Stoughton, and P.-Y. Strub. 2019. "Machine-Checked Proofs for Cryptographic Standards: Indifferentiability of Sponge and Secure High-Assurance Implementations of SHA-3." In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security,* (*CCS*).

Conrado, Giovanna Kobus, Amir Kafshdar Goharshady, and Chun Kit Lam. 2023. "The Bounded Pathwidth of Control-Flow Graphs." *Proc. ACM Program. Lang.* 7 (OOPSLA2). https://doi.org/10.1145/3622807.

Lozano, Roberto Castañeda, Mats Carlsson, Gabriel Hjort Blindell, and Christian Schulte. 2019. "Combinatorial Register Allocation and Instruction Scheduling." *ACM Trans. Program. Lang. Syst.* 41 (3). https://doi.org/10.1145/3332373.

Thorup, Mikkel. 1998. "All Structured Programs Have Small Tree-Width and Good Register Allocation." *Inf. Comput.* 142 (2): 159–81. https://doi.org/10.1006/INCO.1997.2697.