Sébastien Michelland

Ph.D student in computer science

 +33 7 80 51 26 33 •   sebastien.michelland@protonmail.com
 https://silent-tower.net/ • Nov. 1999

Education

École Normale Supérieure de Lyon   **Lyon, France**
Computer Science MSc   *2017–2022*
Graduated BSc in 2018 and MSc in 2020 (both *summa cum laude*). The curriculum covers general computer science theory along with research experience; I focused on languages, semantics and compilers.

Lycée du Parc   **Lyon, France**
Classe préparatoire   *2015–2017*
A two-year intensive training preparing students for the entrance exams to competitive French universities.

Experience

Research

Basic analysis of heap allocation algorithms   *2016–2017*
I implemented and tested several linked-list-based heap allocation algorithms to compare their memory usage and fragmentation to theoretical expectations [2].

Design of a memory-light CPU architecture   *2018*
In architecture class, we experimented with an ISA design that keeps memory bandwidth to a minimum [1]. I also wrote an interactive emulator/debugger and a CHIP-8 emulator in its assembler (repository).

Coq formalization of the dancing links algorithm   *June–July 2018 (Paris, France)*
I proved the dancing links algorithm in Coq and partially proved an accompanying OCaml version [3].

Study of interactions between LLVM passes   *April–June 2019 (Montreal, Canada)*
I collected a database of LLVM optimization passes and dependencies, and sketched software engineering tools that study how they interact to guide empirical phase ordering [4].

Extensions of the congruence closure algorithm   *February–June 2020 (Grenoble, France)*
I developed extensions to the congruence closure algorithm and implemented the resulting decision procedure in OCaml, to be used in an updated version of the Coq tactic congruence [5].

Abstract semantics for monadic interpreters   *October–December 2021 (Lyon, France)*
I designed and implemented in Coq a basic framework for deriving and proving abstract semantics on *interaction trees*. This works makes it possible to extract abstract interpreters for languages defined by interaction trees.

Composable formal semantics for MLIR   *March–July 2022 (Edinburgh, United Kingdom)*
I applied my experience with interaction trees to MLIR, a modular framework for creating LLVM-like compiler IRs. By providing a suitably modular semantics, I enabled basic reasoning on peephole rewrites in MLIR.

Ph.D: Compilation for fault tolerance   *September 2022–ongoing*
My works as a Ph.D student focuses on getting counter-measures for fault injection and side-channels attacks up the compilation chain, by making the compiler generate fault-resistant code from source annotations.

Projects

Unikernel development on embedded calculators *2015–ongoing*

I wrote a unikernel for CASIO graphing calculator applications, which embeds a basic hypervisor that dynamically gets and yields hardware control from/to the official OS. The unikernel provides applications with custom drivers, interrupt-based asynchronous I/O, basic USB 2.0 support, and a custom C99 libc. Undocumented hardware modules were reverse-engineered from the official OS binary. More details here.

Languages and main skills

French: Native speaker

English: CEFR level C2 *CAE 203/210 (2018), TOEFL 116/120 (2021)*

C and embedded development: Advanced *User/kernel land, LLVM internals, Linking, Basic RE*

Functional programming and type theory: Very comfortable *Coq, Lean 4, Haskell*

Also: some experience with C++, Linux administration, LaTeX

References

[1] Florent de Dinechin, Maxime Darrin, Antonin Dudermel, Sébastien Michelland, and Alban Reynaud. Une architecture minimisant les échanges entre processeur et mémoire. In *ComPAS 2018 - Conférence d'informatique en Parallélisme, Architecture et Système*, pages 1–8, Toulouse, France, July 2018.

[2] Sébastien Michelland. Rapport TIPE : Gestion de mémoire informatique. `https://silent-tower.net/static/tipe_heap_allocation.pdf`, Jun 2017.

[3] Sébastien Michelland. Rapport de stage : Permutations et liens dansants vérifiés en CFML. `https://silent-tower.net/static/internship_dancing_links.pdf`, Jul 2018.

[4] Sébastien Michelland. Rapport de stage : exploration et cartographie des passes de LLVM. `https://silent-tower.net/static/internship_llvm_passes.pdf`, Jul 2019.

[5] Sébastien Michelland. Rapport de stage : Une procédure de décision pour relations d'équivalence. `https://silent-tower.net/static/internship_congruence_closure.pdf`, Jun 2020.

Updated May 30th, 2023