Lund, Sweden
✱ *1988*
+46 735 543 963
alexdura@posteo.net
alexdura.github.io

Alexandru Dura

Education

2018–2025 **Doctor of Technology**, *Lund University*, Lund, Sweden
Thesis: Fully Declarative Specification of Static Code Checkers

2011–2012, 2018 **Master's Degree in Computer Science**, *West University*, Timisoara, Romania
Thesis: Adaptive Framework for 3D Computer Vision on IoT Devices

2007–2011 **Bachelor's Degree in Computer Engineering**, *Politehnica University*, Timisoara, Romania
Thesis: Code Generation Using the LLVM Framework

Experience

2018–2025 **Doctoral Student in Software Technology**, *Lund University*, Lund, Sweden
Research in declarative methods for static program analysis.
- Definition, implementation and evaluation of declarative static code checker frameworks targeting Datalog, C and Java
- Representation of syntactic patterns and programs in Datalog
- Program fact extraction and provenance tracking
- Analysis and transformation of Datalog programs
- Automatic incrementalization of static code checkers
- Parsing with ambiguous grammars; general context free parsers
- Implementation of a Datalog engine, using semi-naive evaluation, in Java
- Implementation of Datalog-on-semirings
- Integration of Datalog engines (Soufflé), compiler infrastructures (ExtendJ, Clang) and database systems (sqlite)
- Static analysis for the GObject type system

2022 **Researcher**, *Ericsson*, Lund, Sweden
Research internship in the Standards and Technology group (3 months).
- Static program analyses for speeding up dynamic analyses
- Prototype of a static analysis tool for C programs

2012–2018 **Senior Compiler Engineer**, *Arm*, Lund, Sweden/Trondheim, Norway
Contributor to the Arm Mali compiler.
- Machine code generation for Arm Mali GPUs
- SSA-based optimizations (e.g. swizzle propagation, constant folding, instance-invariant code extraction and evaluation, peephole optimizations)
- Architecture of the GLSL ES 3.2 language extension
- Compiler performance: identified, prototyped and implemented optimisations in all phases of compilation, including instruction selection and scheduling, transformations on the intermediate representation and in the language front-end

2010–2012 **Research Engineer**, *Movidius/Politehnica University*, Timisoara, Romania
Member of the POS CCE Falx Daciae research program.
- Evaluation of multiple compiler frameworks for the development o a custom code generator
- LLVM backend targeting the Movidius SHAVE architecture
- Machine model for an exposed pipeline VLIW architecture
- Vectorizer exploiting super-word level parallelism (SLP)

2008–2010 **Embedded Software Developer**, *Continental Automotive*, Timisoara, Romania
Maintainer of tools for debugging and validation of automotive embedded solutions (part-time position).
- Emulation and message translation of a Controller Area Network (CAN)
- CAN diagnosis environments for driver assistance projects
- Modeling of existing software; auto-generation of software components from UML models

Teaching Experience

2019-2024 Supervision of five master thesis projects, in an academic setting and in collaboration with industry

2018-2025 Teaching Assistant: Program Analysis, Compilers, Concepts of Programming Languages, Functional Programming, Operating Systems, Databases

Skills and Courses

Areas Static Program Analysis, Compiler Construction, Programming Language Design

Programming Languages C, C++, Java, Python, Haskell, Datalog, Prolog

Tools git, bash, docker, LLVM, valgrind, gdb

Coursework Program Analysis, Constraint Programming, Numerical Methods, Machine Learning, Computer Vision, Metaheuristics, Automated Theorem Proving, Digital Signal Processing, Security of Computer Systems, Embedded Systems, Computer Architecture

Natural Languages English (full professional proficiency), Swedish (working proficiency), Romanian (native proficiency)

Projects

MetaDL A Datalog language extension for source-level program analysis and tools for generating the language extension from a description of the analyzed language.

JavaDL A source-level, fully declarative, static code checker framework for Java with incremental evaluation support.

Clog A fully declarative static code checking framework for C

Publications

Alexandru Dura. *Fully Declarative Specification of Static Code Checkers*. Doctoral thesis (compilation), Faculty of Engineering, LTH, Lund University, April 2025.

Alexandru Dura and Christoph Reichenbach. Clog: A declarative language for c static code checkers. In *Proceedings of the 33rd ACM SIGPLAN International Conference on Compiler Construction*, CC 2024.

Alexandru Dura, Christoph Reichenbach, and Emma Söderberg. JavaDL: Automatically incrementalizing java bug pattern detection. *Proceedings of the ACM on Programming Languages*, 5(OOPSLA), 2021.

Alexandru Dura and Hampus Balldin. MetaDL: Declarative program analysis for the masses. In *Proceedings Companion of the 2019 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*, SPLASH Companion 2019.

Alexandru Dura, Hampus Balldin, and Christoph Reichenbach. MetaDL: Analysing Datalog in Datalog. In *Proceedings of the 8th ACM SIGPLAN International Workshop on State Of the Art in Program Analysis*, SOAP 2019.