Mihai Nicola

551-554-1663 | lnicola@stevens.edu | linkedin.com/in/mihainicola | github.com/mihai-nicola

Research Interests

My research aims to advance software reliability. In my work I apply formal methods to develop new techniques for reasoning about program correctness. I focus in particular on programming languages, automated verification, and concurrency.

Education

Stevens Institute of Technology — Hoboken, NJ
Ph.D. in Computer Science — *Sept 2021 - Present*

Politehnica University — Bucharest, Romania
M.S. in Computer Science — *Sept 2018 - Sept 2020*

Faculty of Cybernetics, ASE — Bucharest, Romania
B.S. in Economic Informatics — *Sept 2000 - Jun 2005*

Publications

Nicola, M., Enea, C., & Koskinen, E *(under submission)*. Induction-Reduction Refinement for Concurrent Object Proofs.

Nicola, M., Agarwal, C., Koskinen, E., & Wies, T. (2025). Abstract Interpretation of Temporal Safety Effects of Higher Order Programs. Proceedings of the ACM on Programming Languages, 9(OOPSLA2), 2511-2539.

Chen, A., Fathololumi, P., **Nicola, M.**, Pincus, J., Brennan, T., & Koskinen, E. (2023, October). Better predicates and heuristics for improved commutativity synthesis. In International Symposium on Automated Technology for Verification and Analysis (pp. 93-113). Cham: Springer Nature Switzerland.

Awards & Honors

Provost Doctoral Fellowship. Stevens Institute of Technology (2021)

Talks

Automated Refinement of Concurrent Object Quotients. VDS 2025
Inferring Dependent Types and Effects through Abstract Interpretation NJPLS May 2024

Teaching experience

Teaching Assistant — Stevens Institute of Technology

Compiler Design. Spring 2024, Spring 2025
Concurrent Programming. Fall 2025

Projects

Mechanization of Correctness Proofs for Concurrent Objects in LEAN

Ongoing work exploring the mechanization and automation of safety proofs for lock-free concurrent objects.

Developing a compositional proof framework that derives trace quotients by combining partial-order reduction and Lipton's reduction via induction-reduction tuples capturing recurrent nonblocking synchronization patterns.

Demonstrating expressiveness by verifying canonical lock-free data structures, including a concurrent counter, Treiber Stack, Michael-Scott Queue, Elimination Stack.

evDrift: Static Analysis of Dependent Temporal Safety Effects of Functional Programs

Developed a scalable static analysis for automated verification of temporal safety properties over effect traces for a subset of OCaml, equipped with an ev expression that emits events as side effects

Built an automata-based abstract interpreter for the trace extension operator ev, integrating it into the abstract-interpretation-based dataflow refinement type inference tool Drift

Demonstrated improved scalability and performance compared to state-of-the-art automated safety verifiers

Servois2: Synthesis of commutativity conditions of data structure methods and code

Developed a model-counting-guided heuristic for predicate selection during synthesis, providing quantitative coverage criteria over the search space

Supported rich logical fragments, including linear integer arithmetic (LIA) over bounded domains, and theories of strings, Booleans and arrays.

Professional Experience

Domain Software Architect 2019 – 2021

ING Bank *Bucharest, Romania*

Design and validation of reliability-critical systems, focusing on correctness, and guiding development teams to ensure safety under availability constraints.

Consultant / Senior Software Engineer 2006-2019

United Nations Conference on Trade and Development *Geneva, Switzerland*

Led the design and implementation of configurable, large-scale software systems for international trade and e-government platforms, deployed in more than 30 countries

Additional industry positions omitted for brevity and available upon request.