

AARON ELINE

Washington, DC

443.890.0151 ◊ aaroneline@icloud.com ◊ aaronjeline.github.io

ABOUT ME

Computer Science researcher focused on the areas of programming languages, fuzzing, automated reasoning, and security. Interested in making computer systems orders of magnitude more trustworthy.

EDUCATION

University of Maryland, College Park

2021 - 2022

MS Computer Science.

University of Maryland, College Park

2017 - 2021

BS Computer Science, concentration in Cyber Security.

WORK EXPERIENCE

Amazon Web Services - Automated Reasoning Group

2024-Present

Applied Scientist

- Lead the design and implementation of a random property testing workflow (Launched November 2025) into the Kiro IDE for specification based development.
 - Provides evidence for customers that LLM-synthesized programs conform to specifications.
 - Provides an interactive workflow for refining specifications.
- Design and implementation a novel JVM code reasoning platform.
 - Responsible for implementation of random testing component.
 - Built automated fuzz harness infrastructure
- Managed intern research projects:
 - Automated synthesis of random generators from inductive specifications of the target space in the Lean theorem prover
 - Utilizing language models to drive test case generation for automated disproof of program properties

Amazon Web Services - Automated Reasoning In Identity

2022 - 2024

Applied Scientist

- Designed and implemented the Cedar Policy Language, an open source authorization policy language.
 - Language supports a sound and complete analysis via compilation to SMT, and a sound type system.
 - Performed a formal verification of Cedar in the Lean theorem prover.
 - Built an automated random differential testing setup and ran experiments on effectiveness.
 - Resulted in two research papers appearing at major PL/SE venues.
- Managed an intern research project investigating the effectiveness of various random testing strategies.

Correct Computation

2019 - 2022

Software Engineer / Researcher

- Created *Checked C* language tooling, an extension to the C language that adds spatial memory safety.
 - Developed 3C, a tool for automated best-effort conversion of legacy C code into Checked C, built on top of the Clang compiler infrastructure.

- Developed 5C, a tool that integrates 3C into a develop-in-the-loop workflow for iteratively converting C into Checked C
- Published a research paper at a top PL venue, awarded Distinguished Paper.

PUBLICATIONS

Eline, Aaron. “Does your code match your spec?”. Blog post for the Kiro IDE available at kiro.dev/blog/property-based-testing

Disselkoen, Craig, et al. “How we built Cedar: A verification-guided approach.” Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. 2024.

Cutler, Joseph W., et al. “Cedar: A New Language for Expressive, Fast, Safe, and Analyzable Authorization.” Proceedings of the ACM on Programming Languages 8(OOPSLA1) (2024): 670-697.

Paraskevopoulou, Z., Eline, A., & Lampropoulos, L. (2022, June). Computing correctly with inductive relations. In Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation (pp. 966-980).

Machiry, A., Kastner, J., McCutchen, M., Eline, A., Headley, K., & Hicks, M. (2022). C to Checked C by 3C. Proceedings of the ACM on Programming Languages, 6(OOPSLA1), 1-29. (Distinguished Paper)

TALKS

DC Systems - Testing and Verification : Scalable High Assurance - 2024, watch here

JHU Undergraduate Security and Privacy - Guest Lecturer : Intro to High Assurance Programming - 2024

UPenn Distributed Systems Group - Testing and Verification : Scalable High Assurance - 2024

FSE 2024 - How We Built Cedar: A Verification Guided Approach - 2024

NJPLSS - Cedar: A language for expressing fast, safe, and fine-grained authorization policies - 2024

JHU Undergraduate Security and Privacy - Guest Lecturer - The Cedar Authorization Language - 2023

JHU Graduate Security Group - Gradually Achieving Memory Safety - 2022