Anastasiya Kravchuk-Kirilyuk

Education

Harvard University May 2026
Ph.D. Candidate in Computer Science (Programming Languages)

Honors and Awards: Siebel Scholar (2026), Certificate of Distinction in Teaching (2022)

Princeton University June 2020
M.S.E. in Computer Science

University of Pennsylvania May 2017
B.S.E. in Computer Science, Minor in Mathematics

Honors and Awards: Dean's List (2016-2017 AY), CIS Faculty Appreciation Award (2017)

Skills

Programming Languages: Scala, Java, Python, C#, OCaml, R (Intermediate).

Verification and Tooling: Rocq Theorem Prover, Dafny, Verified Software Toolchain, Boogie (Intermediate).

Research: Type systems, formal verification, modularity, language design, proof evolution, LLM-aided synthesis.

Professional Experience

Applied Scientist Intern, AWS – Boston, MA Summer 2024, Summer 2023

- Worked with the Dafny team at ARG (Automated Reasoning Group) to improve long-term maintainability of critical systems by verifying backward compatibility of code commits. Additionally, explored extensions for the module refinement feature in Dafny to support proof evolution.
- Returned for a second internship to develop advanced features for a cross-language equivalence checker, automating comparison predicate generation for complex heap-based data types (e.g., structs and arrays), and driving scalable, multi-language program verification.

Research Experience

Doctoral Researcher, Harvard University September 2020 – Present
Supervised by Nada Amin Cambridge, MA

- Designed and implemented a family polymorphic calculus (Persimmon), enabling extensible variant types, pattern matching, and modular code reuse through nested family polymorphism. Published this work in a top-tier venue (OOPSLA 2024), advancing the state of formal methods for scalable real-world systems.
- Prototyped and evaluated LLM-driven tools for synthesizing verified code, integrating formal methods with AI to guarantee correctness of generated code.
- Made substantial contributions to successful grant proposals securing ∼$480,000 in competitive funding, including an NSF Award (∼$400K, 2 years) and an Amazon Research Award ($80K, 1 year).
- Invited as expert speaker at OOPSLA 2024, NEPLS 2025, and TYPES 2021, presenting research on family polymorphism and proof extensibility with direct applications to modular design and scalable verification.

Graduate Researcher, Princeton University September 2018 – June 2020
Supervised by Andrew Appel Princeton, NJ

- Defined a general API for ordered database indexes, creating a reusable abstraction for formally verifying data structure implementations.
- Formally verified a B+-tree with cursors against this API using Rocq, the Verified Software Toolchain (VST), and CompCert, ensuring key safety and correctness properties through mechanized proofs.
- Packaged the verified B+-tree as a modular, formally verified C component – a Verified Software Unit, encapsulating correctness guarantees behind a clean interface for modular reuse and system-level integration.
- Invited to present at CoqPL 2021, highlighting this work as a model for building reusable, verifiably correct infrastructure components.

Research Assistant, University of Pennsylvania June 2017 – June 2018
Supervised by Stephanie Weirich Philadelphia, PA

- Implemented and proved correctness of eta-equivalence rules for dependently typed Haskell as part of the Corespec project, supporting deeper reasoning over Haskell programs. Fully mechanized the extension in Rocq, using Ott and LNgen to generate abstract syntax and variable binding infrastructure.
- Co-authored a peer-reviewed publication on extending Dependent Haskell with eta-equivalence (TYPES 2019), advancing the theory of dependently typed functional programming. Invited to co-present work on scalable proof techniques at CoqPL 2018, showcasing practical approaches to mechanized proof engineering.

Leadership and Project Management

Harvard University, Teaching Fellow 2021 – 2026
Courses: Programming Languages, Advanced Topics in Programming Languages, Formal Methods for Security

- Supported course delivery for graduate and undergraduate classes of 30+ students in programming languages and formal methods, consistently earning excellent student reviews and satisfaction ratings (4.8/5 on average).
- Collaborated closely with instructors to structure assignments, design lecture materials, and provide detailed feedback, contributing to high-quality course experiences.
- Led sections and office hours, managed homework materials and grading, demonstrating strong communication, organizational, and leadership skills in fast-paced academic environments.
- Guided students through complex content, including cutting-edge research topics, reinforcing deep technical understanding and improving student engagement.

Princeton University, Assistant Instructor 2018 – 2020
Courses: Introduction to Computer Science

- Led four weekly sections and twice-weekly office hours for a 200-student undergraduate course, collaborating with 10+ co-instructors to coordinate exam preparation, grading, and student support.

University of Pennsylvania, Teaching Assistant 2014 – 2017
Courses: Introduction to Computer Science, Software Development, Programming Paradigms

- As head teaching assistant, led and scaled a 20+ member TA team for a 200+ student course, hiring and training new staff, creating onboarding materials, scheduling coverage, and running weekly staff meetings to ensure consistent, high-quality instruction.
- Coordinated and managed software development projects in educational settings, applying project management methodologies to facilitate collaboration and timely delivery.

Publications

Anastasiya Kravchuk-Kirilyuk, Fernanda Graciolli, and Nada Amin. **"The Modular Imperative: Rethinking LLMs for Maintainable Software."** Proceedings of the 1st ACM SIGPLAN International Workshop on Language Models and Programming Languages. 2025.

Anastasiya Kravchuk-Kirilyuk, Gary Feng, Jonas Iskander, Yizhou Zhang, and Nada Amin. **"Persimmon: Nested Family Polymorphism with Extensible Variant Types."** Proceedings of the ACM on Programming Languages 8.OOPSLA1 (2024): 698-724.

Anastasiya Kravchuk-Kirilyuk, Antoine Voizard, and Stephanie Weirich. **"Eta-Equivalence in Core Dependent Haskell."** 25th International Conference on Types for Proofs and Programs (TYPES 2019). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Professional Service

- Program Committee Member, The Scala Workshop 2026.
- Artifact Evaluation Committee Member, OOPSLA 2025 (Distinguished Artifact Reviewer Award).
- Harvard SEAS Undergraduate Research Open House. Met with undergraduate students to encourage research access for underrepresented groups (2021-2023).
- Taught introduction to verification lectures at MIT Spark (Spring 2021) and MIT Splash (Fall 2020).
- Formal Logic and Software Verification using Interactive Theorem Provers. Co-hosted workshop at the ACM Philadelphia Region Celebration of Women in Computing Conference (Spring 2018).