ANFISA BOGDANENKO

UPDATED NOV 2026

MSc in Computer Science, Northeastern University, 2026
fisafisa.net · an@fisafisa.net · github.com/fisafisafisa

ABOUT

I am interested in reasoning about properties of higher-order programs and then exploiting them. I am especially drawn to type-and-effect systems and abstract interpretation, and I really enjoy compiler optimization tech. I also love DSLs because I believe that expressive abstraction design empowers human reasoning and problem-solving.

INTERESTS functional languages, program analysis, effects, optimization, DSLs

EDUCATION

MSc in Computer Science

JAN – AUG 2026

Northeastern University · Boston, MA, USA

- GPA: 4.00 / 4.00

THESIS *SIMD/MIMD Semantics of a Higher-Order Rank-Polymorphic Programming Language* · pdf · slides
ADVISOR Prof. Olin Shivers

BSc in Computer Engineering + Minor in Mathematics

SEP 2021 – DEC 2025

Northeastern University · Boston, MA, USA

- GPA: 3.99 / 4.00
- International Scholar Merit Scholarship
- PEAK Summit Fall 2025, Undergraduate Research Fellowship

CAPSTONE *SoundLab: Analog Subtractive Synth* · poster
ADVISOR Prof. Bahram Shafai

PUBLICATIONS

Lackadaisical Contracts for Higher-Order Functions

POPL 2027

Cameron Moy, Anfisa Bogdanenko, Matthias Felleisen · pdf · code 1 · code 2

EXPERIENCE

Student Researcher

SEP 2025 – AUG 2026

Programming Research Laboratory @ NEU · Boston, MA, USA

Research on Remora, a DARPA-funded parallel array language for scientific and AI code, supervised by Prof. Olin Shivers. This work became my MSc thesis.

- Developed a formal semantics for a CPS IR, which is used to derive a static analysis identifying SIMD fragments in MIMD programs
- Prepared presentations for regular DARPA events

Teaching Assistant

JAN – APR 2025

Khoury College @ NEU · Boston, MA, USA

Assisted Prof. Steven Holtzen with CS4400 Programming Languages.

- Helped design homework assignments and quizzes
- Held weekly office hours and graded exams

Research Assistant JUL – DEC 2024

Programming Research Laboratory @ NEU · Boston, MA, USA

Six-month research co-op under supervision of Prof. Olin Shivers. Worked with Vadym Matviichuk on the implementation of Remora, a parallel array language.

- Enhanced the compiler pipeline (OCaml, CUDA C++)
- Implemented a new 2D convolution primitive, yielding a 70% performance boost
- Fixed bugs in the interpreter (Racket) for a typed variant of Remora

Electro-Optical Testing Engineer JUL – DEC 2023

E Ink Innovation Center · Billerica, MA, USA

Six-month co-op at E Ink, the maker of electronic paper displays.

- Developed firmware (C, FreeRTOS) for a custom test controller and documented its features
- Automated data collection for a previously manual measurement, saving about 160 hours of testing a year
- Ran routine experiments in the electro-optical lab and presented results at weekly team meetings

Part-time Research Assistant JAN – JUL 2023

Programming Research Laboratory @ NEU · Boston, MA, USA

A study of program synthesis tools versus LLMs, supervised by Prof. Arjun Guha and John Gouwar.

- Evaluated program synthesis tools (e.g. Myth, Trio) on the HumanEval benchmarks
- Translated the benchmarks into each tool's OCaml-based input and ran the tools in Docker

PROJECTS

CPS compiler for Scheme SPRING 2026

CS6983 Compiling Functional Languages

Implemented a CPS compiler in Nanopass (Racket) from a subset of Scheme to abstract machine instructions. Besides many simplification passes, it performs control-flow analysis to properly convert Scheme closures to machine code.

Explainer on categorical semantics of intuitionistic linear logic FALL 2025

CS7480 Special Topics: Categories for PL · pdf

As a final project, students prepared "explainers" that gave categorical semantics of a chosen formal system. My group's write-up focused on intuitionistic linear logic, to better understand affine type systems.

Compiler from a Python-like language to x86 SPRING 2025

CS6410 Compilers

Implemented a compiler (OCaml, C) for a dynamically typed functional language with a copying garbage collector, tail calls, dead-variable elimination, register allocation, and call-with-current-continuation.

scsh, a Unix shell embedded in Scheme FALL 2024

Maintainer of the open-source project · code

Ported the last scsh release (2006) to the latest stable Scheme48, modernized its interface (adding Unicode support, dropping outdated features like file locking), and rewrote the manual in Scribble.